Entwicklung eines integrierten Softwarepaketes zur Unterstützung des Designs und der Synthese artifizieller Gene in einer Hochdurchsatz-Umgebung

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

der Naturwissenschaftlichen Fakultät IV
- Chemie und Pharmazie -
der Universität Regensburg

vorgelegt von

David Raab
aus Regensburg
2003

Promotionsgesuch eingereicht am: 29.09.2003

Tag des Kolloquiums: 03.11.2003

Die Arbeit wurde angeleitet von: PD Dr. J. Enderlein

Prüfungsausschuß:

1. Gutachter: PD Dr. J. Enderlein

2. Gutachter: Prof. Dr. R. Wagner

3. Prüfer: Prof. Dr. O. Wolfbeis

Zusammenfassung

Zusammenfassung

Synthetische Gene, aus Oligonukleotiden aufgebaute und bis zu mehrere Kilobasen große dsDNA-Fragmente, haben sich bereits in vielen Bereichen der Molekularbiologie und der molekularen Medizin zu einem wertvollen Werkzeug entwickelt. Dennoch wird das hohe Potential dieser Technologie bislang nur zu einem Bruchteil genutzt. Dies liegt vor allem darin begründet, dass trotz enormer Fortschritte der Aufbau eines synthetischen Gens für den einzelnen Forscher immer noch mühsam und kostspielig ist. Darüber hinaus werden die Möglichkeiten des rationalen Sequenzdesigns mangels geeigneter Software bis jetzt nur unzureichend genutzt. Die Idee liegt jedoch nahe, durch hochgradige Automatisierung des Gensyntheseprozesses eine starke Reduktion sowohl der Synthesekosten als auch der Dauer zu erreichen. Um diesem Ziel näher zu kommen, wurde im Rahmen dieser Arbeit ein integriertes Softwarepaket erstellt, welches die Hochdurchsatzsynthese artifizieller Gene angefangen beim Design über die Syntheseplanung bis zur Herstellung und Qualitätskontrolle unterstützt.

Kernstück der GeneOptimizer-Software bildet ein benutzerfreundlicher Sequenzeditor, der alle üblichen Bearbeitungsfunktionen beherrscht und das „Reißbrett" für das Sequenzdesign darstellt. Um die (codierende) Sequenz des synthetischen Genes optimal auf den Einsatzzweck und das Expressionssytem anzupassen, müssen häufig mehrere Kriterien, wie Anpassung der Kodonwahl und des GC-Gehaltes, Ausschluss von bestimmten DNA-Motiven und Repetitionen, Vermeidung stabiler Sekundärstrukturen etc. berücksichtigt werden. Durch die Degeneriertheit des genetischen Codes kann allerdings bereits ein sehr kleines Protein durch eine Vielzahl verschiedener DNA-Sequenzen codiert werden. Um eine Sequenz aufzufinden, welche den o.g. Kriterien in möglichst idealer Weise entspricht, wurde ein Algorithmus entwickelt, welcher auf der Verwendung eines „gleitenden Kombinationsfensters" beruht. Innerhalb dieses mehrere Kodons langen Fensters werden alle möglichen Kombinationen synonymer Kodons generiert und jede erzeugte Kombination zusammen mit der bereits optimierten stromaufwärts gelegenen Sequenz durch eine Gütefunktion mit anwendergewichteten Parametern bewertet. Das erste Kodon der optimalen Kombination wird festgelegt und das Kombinationsfenster um eine Aminosäure in 3'-Richtung weitergeschoben. Der Prozess wird so lange wiederholt, bis die gesamte optimierte DNA-Sequenz ermittelt wurde.

Zusammenfassung

Die Software bietet dem Anwender komplementär zu den Optimierungsoptionen eine Vielzahl von Funktionen zur Sequenzanalyse und Sequenzannotation, so dass der Anwender in einem Optimierung-Analyse Feedback-Prozess die Gewichtung der einzelnen Optimierungskriterien so lange verändern kann, bis die Sequenz seinen Vorstellungen entspricht.

Anschließend unterstützt das Programm den Anwender bei der optimalen Unterteilung der Gesamtsequenz in Subfragmente, dem Anfügen optionaler Linkersequenzen und schließlich der Generierung der Oligonucleotidsequenzen. Dabei werden die Oligolängen auf eine bestimmte Schmelztemperatur abgestimmt und der Anwender ebenso vor möglichen Syntheseproblemen durch Fehlhybridisierungen gewarnt.

Das Softwarepaket unterstützt auch den Produktionsprozess, indem die Sequenzen von Subfragmentklonen automatisch auf ihre Übereinstimmung mit der Vorgabesequenz hin bewertet werden. Durch die Klassifizierung von Sequenzfehlern in „sichere Fehler" und „unwahrscheinliche Fehler" (durch fehlerhaftes oder unsicheres Basecalling erzeugt) kann auch bei qualitativ schlechten Sequenzierungsläufen eine hohe Bewertungssicherheit erreicht werden.

Auch bei der finalen Qualitätskontrolle der Gesamtsequenz wirkt die Software durch Alignment- und annotationsfunktionen unterstützend mit.

Mittlerweile wurden über tausend synthetische Gene mit Hilfe der GeneOptimizer Suite entworfen und/oder ihre Herstellung softwareseitig unterstützt.

Inhaltsverzeichnis

A Einführung

A.1 Einleitung

Die vollständig chemische Synthese von Oligodesoxyribonukleotiden gehört zweifellos zu den Entwicklungen, die die Fortentwicklung in der Molekularbiologie mit am stärksten beeinflusst haben. Elementare molekularbiologische Techniken wie PCR, ortsgerichtete Mutagenese und DNA-Sequenzierung sind ohne diese kurzen DNA-Bausteine mit genau definierter Sequenz kaum vorstellbar.

Während die Synthese von Oligonukleotiden in den 70er Jahren noch wenigen Laboratorien vorbehalten war, können diese dank ausgefeilter organischer Syntheseschritte und weit fortgeschrittener Automatisierung heute via Internet von spezialisierten Dienstleistern zu einem Basenpreis von weit unter 1 € bestellt und bereits am nächsten Tag eingesetzt werden.

Dennoch ist die problemlos erreichbare Länge von Oligonukleotiden auf einige Dutzend Basen beschränkt. Bereits Ende der 60iger Jahren wurde deshalb der Versuch unternommen, durch das Zusammenfügen von Oligonukleotiden längere doppelsträngige DNA-Abschnitte zu erhalten. Dieses Bestreben mündete schließlich nach Aufklärung des genetischen Codes 1970 in die Synthese des ersten synthetischen Gens durch Khorana und Mitarbeiter [Khorana 1970].

Während hier zunächst das „proof of principle" im Vordergrund stand (exakter Nachbau einer natürlichen DNA-Sequenz) wurde bald von der einzigartigen Möglichkeit vollsynthetischer Gene Gebrauch gemacht, die DNA-Sequenz basengenau auf die jeweiligen Anforderungen hin designen zu können. Während bei Kösters Synthese des Angiotensin II noch synthetische Aspekte die Wahl der Kodons bestimmten [Köster 1978], orientierte sich Itakura bei der Synthese des Somatostatin bereits an der Kodonwahl des Phagen M2 [Itakura 1977].

Mittlerweile sind synthetische Gene zu einem wertvollen Werkzeug in fast allen Bereichen der Molekularbiologie und molekularen Medizin geworden. Dennoch wird das hohe Potential dieser Technologie bislang nur zu einem Bruchteil genutzt. Dies liegt vor allem darin begründet, dass trotz enormer Fortschritte der Aufbau eines synthetischen Gens für den einzelnen Forscher immer noch mühsam und kostspielig ist. Darüber hinaus werden die Möglichkeiten des rationalen Sequenzdesigns mangels geeigneter Software bis jetzt nur unzureichend genutzt. Dennoch zeichnet sich eine Entwicklung ab, die der im Bereich der Oligo-Synthese ähnlich ist. Die Idee liegt nahe, durch hochgradige Automatisierung des Gensyntheseprozesses eine starke Reduktion sowohl der Synthesekosten als auch der Dauer zu erreichen. Innerhalb dieser Bestrebungen kommt der Bioinformatik eine entscheidende Rolle zu. Zentrale

Aufgabe ist hier zum einen die Entwicklung eines Algorithmus, der es gestattet, anhand vorgegebener Ziele und Rahmenbedingungen automatisch eine optimale DNA-Sequenz zu ermitteln; zum anderen müssen auf dem Weg von der Sequenz zum fertigen Gen zahlreiche Produktionsschritte softwaretechnisch unterstützt werden.

Können die interdisziplinär zwischen Chemie, Biologie und Informatik/ Automatisierungstechnik angesiedelten Herausforderungen auf dem Weg zur effizienten und kostengünstigen Gensynthese erfolgreich gelöst werden, so ist abzusehen, dass in wenigen Jahren synthetische Gene genauso zum Laboralltag gehören, wie heute schon die Oligonukleotide.

A.2 Herstellung synthetischer Gene

Oligonukleotide im Längenbereich von 40-80 Basen bilden die „Bausteine" der Gensynthese. Für ihre Herstellung hat sich die hochautomatisierte Festphasensynthese, z.B. an *Controlled Pore Glass* als Matrix, etabliert. Dabei wird das Oligonukleotid Base für Base vom 3'-Ende her aufgebaut. Wenngleich über die Herstellung eines kompletten synthetischen Genes durch Synthese zweier sehr langer komplementärer Oligonucleotide berichtet wurde [Ciccarelli 1991], so erweist sich die Darstellung von Oligonukleotiden mit über ca. 100 Basen oft als äußerst schwierig. Die heute fast ausschließlich verwendete β-Cyanoethylphosphoramidit-Synthese in Verbindung mit geeigneten Schutzgruppen für die Heterozyklen zeichnet sich zwar durch eine für organische Reaktionen geradezu sensationelle Spezifität und Effizienz aus, dennoch nimmt die Kopplungseffizienz mit wachsender Kettenlänge ab. Doch auch mit einer durchschnittlichen Ausbeute von 98% pro Kopplung erhält man bei der Synthese eines 100 Basen-Oligonucleotides nur $0.98^{100}*100=13\%$ Volllängenprodukt, welches auch sporadisch Insertionen und Deletionen aufweisen kann. Überdies wird das wachsende Oligonucleotid bei jedem Syntheseschritt erneut hochreaktiven Reagenzien ausgesetzt, wodurch es besonders am 3'-Ende zu verstärkten Basenmodifikationen durch Nebenreaktionen kommen kann. Aus diesen Gründen werden längere doppelsträngige DNA-Fragmente üblicherweise über enzymatische Methoden aus kürzeren Oligonukleotiden aufgebaut. Dabei kann man prinzipiell polymerase- und ligasebasierte Verfahren unterscheiden.

Voraussetzung für den Aufbau durch Ligation ist, dass die zur Verknüpfung bestimmten Oligonukleotide am 5'-Ende entweder chemisch oder enzymatisch phosphoryliert werden.
Die Ligation kann entweder so durchgeführt werden, dass zunächst mehrere kurze Duplex-Stücke durch Hybridisierung zweier über fast ihre ganze Länge komplementärer Oligonucleotide dargestellt werden, welche aber wenige Basen lange einzelsträngige Überhänge aufweisen. Indem die Sequenz der Überhänge so gewählt wird, dass das Ende eines Duplexes definiert nur mit dem Anfang eines bestimmten anderen Duplexes hybridisieren kann, ist es möglich, die Duplexe in vorgegebener Reihenfolge miteinander zu ligieren.
Dadurch wird ein (ggf. bis auf die Enden) durchgehend doppelsträngiges DNA-Fragment erhalten.

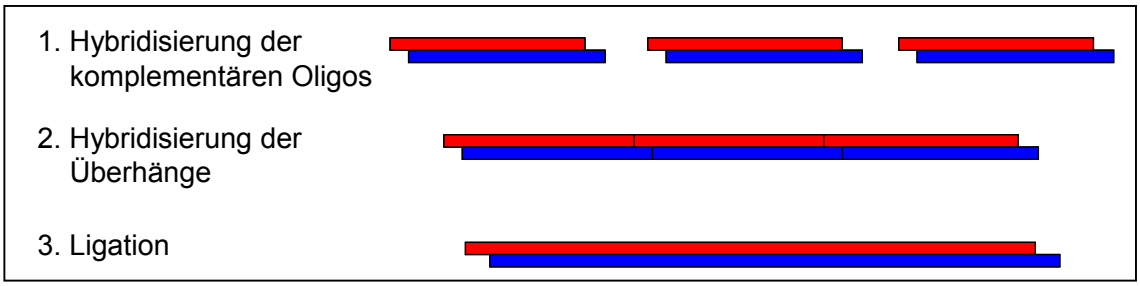


Abb. A.2-1 Herstellung von dsDNA durch Ligation von Fragmenten mit vollständig überlappenden Enden

Zwar müssen bei dieser Methode sowohl der *sense*- als auch der *antisense*-Strang zunächst vollständig als Oligonucleotide synthetisiert werden, die Methode zeichnet sich jedoch durch eine hohe Zuverlässigkeit auch bei schwierigen Sequenzen aus.

Alternativ kann auch nur die *sense*-Strang-Sequenz komplett über Oligonucleotide abgedeckt werden. Zusammengehörende Oligos hybridisieren dabei zunächst mit entgegengesetzten Enden an Fängeroligos und können dann ligiert werden [Chen 1993].

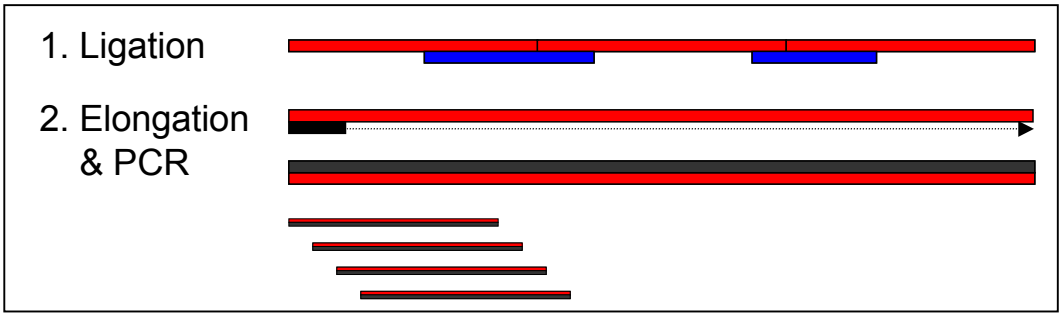


Abb. A.2-2 Herstellung von dsDNA durch Ligation von Oligonukleotiden mit Hilfe von „Fängeroligos"

Beide Methoden können in vielen Fällen als Eintopf-Reaktionen durchgeführt werden, d.h. alle zum Aufbau eines Fragments benötigten Oligonucleotide werden in einer einzigen Ligations-Reaktion eingesetzt. Prinzipiell ist jedoch auch die sequenzielle Durchführung als Festphasenreaktion möglich [Staahl 1993].
Bei beiden Methoden werden die gebildeten Vollängenprodukte in der Regel abschließend mittels PCR aus der Reaktionsmischung hochamplifiziert.

Zur Durchführung einer Polymerase-basierten Gensynthese wird die Sequenz zunächst in kurze Abschnitte von der Länge eines Oligonucleotides unterteilt, welche sich gegenseitig um 15-20 Basen überlappen. Dann werden den Abschnitten entsprechende Oligonucleotide synthetisiert, jedoch abwechselnd mit der Sense und Antisense-Sequenz. In einer Polymerase-Elongationsreaktion können nun jeweils

zwei aufeinanderfolgende Oligonucleotide durch den Überlapp gegenseitig als Primer fungieren [Dillon 2000].

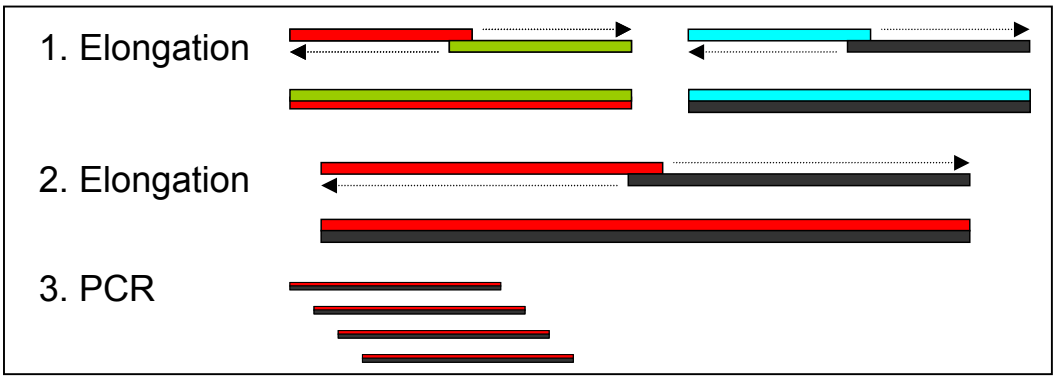


Abb. A.2-3 Herstellung von dsDNA über Primer-Elongation und PCR

Durch wiederholte Denaturierungs-Hybridisierungs- und Elongationsschritte werden in jedem Zyklus längere dsDNA-Stücke aufgebaut, bis schließlich geringe Mengen Volllängenprodukt gebildet werden. Dieses kann mittels PCR zur weiteren Verwendung amplifiziert werden.

Ein völlig neuartiges Verfahren stellt die sogenannte „Sloning"-Technik dar, welche im Prinzip eine enzymatische Festphasensynthese unter Verwendung einer sehr großen Oligo-Bibliothek darstellt. Das Verfahren befindet sich jedoch noch im Entwicklungsstadium [Schatz 2000].

Aufgrund der bereits in den Oligonukleotiden enthaltenen sporadischen Fehler, wie auch durch über die PCR eingeführte Mutationen werden nur wenige der PCR-generierten Stränge über ihre gesamte Länge fehlerfrei sein. Tatsächlich steigt die Zahl der fehlerbehafteten Stränge mit zunehmender Länge der DNA-Fragmente exponentiell an. Aus diesem Grunde werden längere Sequenzen (mehrere kbp lang) zunächst an geeigneten Stellen in kleinere Subfragmente (mehrere hundert bp lang) unterteilt. Diese werden nach einer der oben beschriebenen Methoden als dsDNA hergestellt, in einen Vektor kloniert und ein E.coli Stamm mit diesem transfiziert. Der Transfektionsansatz wird ausplattiert und nach der Inkubation der Platten übernacht wird von mehreren Kolonien jeweils das Vektorinsert mittels PCR hochamplifiziert und sequenziert. Die erhaltenen Sequenzdaten werden dann mit der vorgegebenen Zielsequenz verglichen.
Wird eine Kolonie gefunden, welche ein Insert mit völlig korrekter Sequenz enthält, so kann aus dieser nach einer weiteren Wachstumspause das Plasmid in ausreichender Menge gewonnen werden, um das Insert mit Restriktionsenzymen herauszuschneiden. Die auf diese Weise in klonaler Reinheit gewonnenen

sequenzkorrekten Subfragmente werden anschließend zusammenligiert und kloniert. Nachdem über das oben geschilderte Screening-Verfahren eine Bakterienkolonie mit dem fehlerfreien Gesamtkonstrukt gefunden wurde, kann aus dieser das im Vektor enthaltene fertige synthetische Gen als Plasmidpräparation in größerer Menge gewonnen werden.

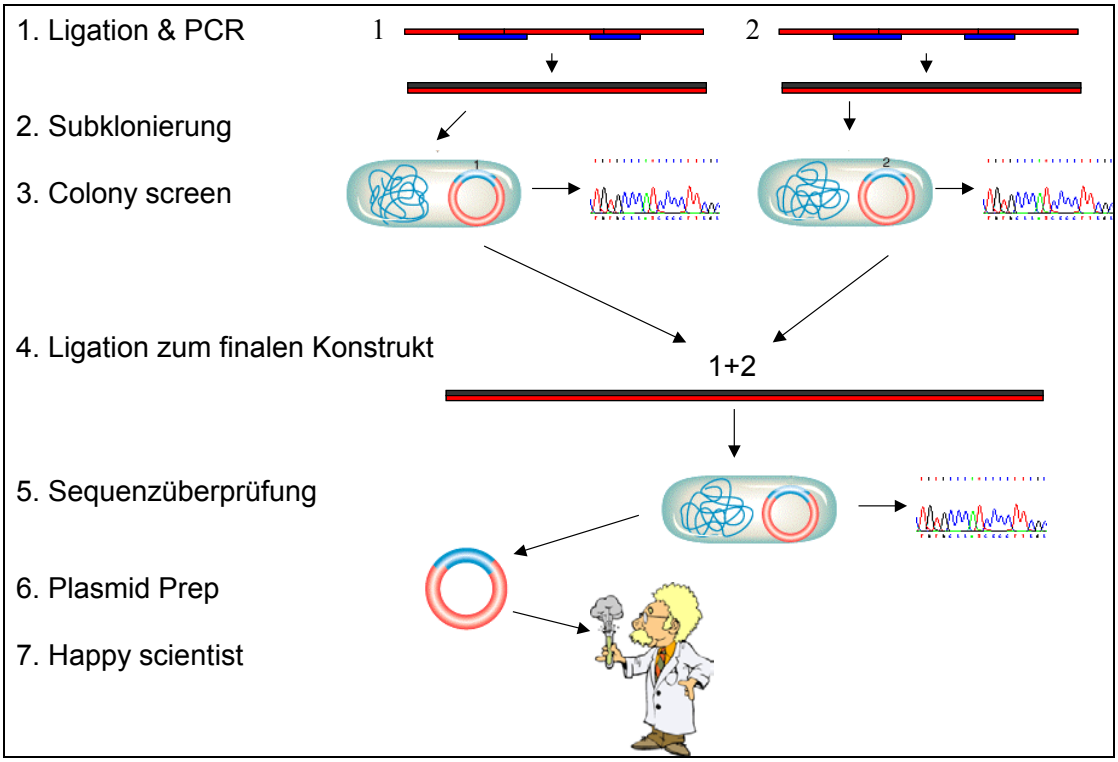


Abb. A.2-3 Schematisch dargestellter Ablauf der Gensynthese

A.3 Einsatzbereiche synthetischer Gene

Synthetische Gene finden vor allem dann Anwendung, wenn die Anpassung einer DNA-Sequenz an die experimentellen Erfordernisse mit konventionellen Methoden, wie ortsgerichtete Mutagenese zu aufwändig wäre oder ein natürliches DNA-Template nur schwer oder gar nicht erhalten werden kann. Synthetische Gene bieten die einzigartige Möglichkeit des „Database Cloning", das heißt auf natürliche DNA als Startmolekül für weitere Modifikationen kann komplett verzichtet werden. Statt dessen kann die Sequenz ohne Rücksicht auf die technischen Limitationen konventioneller Klonierungs- und Mutagenesestrategien optimiert werden.
Aufgrund dieser Vorteile haben synthetische Gene bereits breite Anwendung in vielen Gebieten der Molekularbiologie und Biomedizin gefunden. Exemplarisch werden nachfolgend wichtige Einsatzfelder aufgeführt.

A.3.1 Steigerung der Expression

Während der genetische Code nahezu universelle Gültigkeit besitzt, werden die für eine Aminosäure synonymen Kodons von verschiedenen Spezies im Verhältnis zueinander unterschiedlich häufig genutzt [Smith 1996]. So wird z.B. die Aminosäure Arginin in der Maus zu 21% durch das Kodon AGG codiert, in *Escherichia coli* jedoch nur zu 2%. Dabei besteht eine starke Korrelation zwischen dieser sog. „Codon Usage" und den entsprechenden tRNA-Frequenzen [Ikemura 1982, Dong 1996]. Die Verwendung von nativen Maus-Genen zur heterologen Expression in E.coli wird daher oftmals zu niedrigen Proteinausbeuten führen oder sogar gänzlich versagen, da die Translation des AGG-Kodons aufgrund der niedrigen Konzentration der entsprechenden tRNA nicht effizient vollzogen werden kann. Ebenso kann es beispielsweise durch eine Verschiebung des Leserasters zu Fehltranslationen kommen [Kane 1995]. Bei der Kodon-Optimierung wird daher versucht, jedes Kodon der heterologen DNA durch das Kodon zu ersetzten, welches im Ziel-Organismus (in hochexprimierten Genen) für die entsprechende Aminosäure am häufigsten verwendet wird. Dabei muss allerdings die unbeabsichtigte Generierung expressions-limitierender cis-aktiver Sequenzmotive, wie beispielsweise Splice-Sites oder vorzeitiger Polyadenylierungs-Signalsequenzen etc., vermieden werden. Dies gilt auch für möglicherweise die genetische Stabilität verringernde Sequenzrepetitionen und die potentielle Ausbildung stabiler RNA-Sekundärstrukturen. Ebenso kann je nach Einsatzgebiet die Anpassung des GC-Gehaltes oder die Vermeidung von Homologien zu gegebenen DNA-Sequenzen erwünscht sein. Idealerweise fließen alle

diese Kriterien im Rahmen einer rationalen Sequenzoptimierung bei der Auswahl der synonymen Kodons mit ein.

Aminosäure	Kodon	*Arabidopsis thaliana*	*Homo sapiens*	*Saccharomyces cerevisiae*	*Escherichia coli*
Ala	GCG	0.14	0.11	0.11	0.35
	GCA	0.27	0.23	0.29	0.21
	GCT	0.44	0.26	0.38	0.16
	GCC	0.16	0.40	0.22	0.27
Arg	AGG	0.20	0.20	0.21	0.02
	AGA	0.35	0.20	0.48	0.04
	CGG	0.09	0.21	0.04	0.10
	CGA	0.12	0.11	0.07	0.06
	CGT	0.17	0.08	0.15	0.38
	CGC	0.07	0.19	0.06	0.40
Leu	TTG	0.22	0.13	0.28	0.13
	TTA	0.14	0.07	0.28	0.13
	CTG	0.11	0.40	0.11	0.50
	CTA	0.11	0.08	0.14	0.04
	CTT	0.26	0.13	0.13	0.10
	CTC	0.17	0.20	0.06	0.10

Abb. A.3.1-1 Vergleich der Codon Usage unterschiedlicher Organismen für die Aminosäuren Alanin, Arginin und Leucin (Quelle: Kazusa Codon Usage Database [Nakamura 2000]). Die bei der heterologen Expression in E. coli besonders problematischen Kodons AGG und AGA sind rot gekennzeichnet. Betrachtet man ausschließlich die Codon Usage hochexprimierter Gene, fallen die Unterschiede oft sogar noch drastischer aus.

A.3.2 Kombinatorische Biologie

Während die natürliche Evolution meist in Zeitmaßstäben von vielen Millionen Jahren abläuft, so können unter Ausnutzung geeigneter Mutations- und Selektions– bzw. Screeningmethoden evolutive Prozesse heute „im Reagenzglas" in wenigen Tagen durchgeführt werden. Vorraussetzung dafür sind geeignete Methoden zur Einführung von Mutationen oder zur Randomisierung ganzer Genabschnitte. Klassische Verfahren hierfür sind beispielsweise die Anwendung der „error-prone PCR" oder die Nutzung eines Mutator-*E.coli*-Stammes, welcher sich durch eine Exonuclease-defiziente DNA Polymerase III auszeichnet. Nachteilig ist hierbei jedoch, dass weder die Art noch die Position der eingeführten Mutationen genau festgelegt werden können. Im Gegensatz dazu können während der Synthese (teil)randomisierter Oligonucleotide genau definierte Nucleotidpositionen mit einem festgelegten Verhältnis der 4 Basen randomisiert werden. Diese Oligos können schließlich zum Aufbau eines kompletten synthetischen Gens verwendet werden. Verwendet man während der Oligosynthese Nucleotid-Trimere statt Monomere, ist es sogar möglich, Gene zu synthetisieren, welche an einer definierten Position für eine genau festgelegte Mischung

verschiedener Aminosäuren codieren. Ein bekanntes Beispiel hierfür ist die HuCAL-Antikörperbibliothek der Firma MorphoSys AG [Knappik 2000]. Die CDR („complementarity determinig region") eines Antikörpers umfasst ca. 8 Aminosäuren, so dass theoretisch $2.56*10^{10}$ verschiedene Kombinationen möglich sind. Um mit einer möglichst kleinen Molekülzahl dennoch eine Bibliothek mit möglichst großer struktureller Vielfalt zu erhalten, wurde zunächst die Häufigkeitsverteilung der Aminosäuren auf die acht Positionen in natürlichen Antikörpern bestimmt bzw. die für die Antikörper-Antigen-Erkennung wichtigsten Aminosäuren identifiziert. Indem für die Synthese der CDR-Oligonucleotide definierte Mischungen jeweils für eine bestimmte Aminosäure codierender Trinucleotide verwendet wurden, spiegelten die so generierten CDR's die natürliche Aminosäuren-Häufigkeitsverteilung wieder. Für die ausserhalb der CDR's liegenden Aminosäuresequenzen der V_L und V_H Ketten wurden wenige Consensus-Sequenzen generiert, welche nach Optimierung der DNA-Sequenz auf Expression in *E. coli* ebenfalls vollsynthetisch dargestellt wurden. Die Selektion antigen-spezifischer Antikörper erfolgt dann beispielsweise mit der Phage-Display-Technik.

Im Rahmen der kombinatorischen Biologie können jedoch nicht nur passende Ligand-Rezeptorpaare aufgefunden werden, sondern beispielsweise auch die Eigenschaften von Enzymen zielgerichtet verändert werden. Dabei wird die initiale Randomisierung oft noch mit Rekombinationstechniken, wie dem DNA-Shuffling, kombiniert.

Durch das „Enzym-Engineering" kann etwa die Substratspezifität verändert werden (Umwandlung einer Aspartat-Aminotransferase in eine effiziente Valin-Aminotransferase [Taylor 2001]), oder die Thermostabilität erhöht werden (Entwicklung einer Thermolysin-artigen Protease mit einem um 21°C erhöhten Temperaturoptimum (funktionell bei 100 °C) ohne Verringerung der enzymatischen Aktivität bei 37°C [Arnold 1998]).

A.3.3 DNA-Vakzine

Seit der Beobachtung von Wolff et al. [Wolff 1990], dass die intramuskuläre Injektion eines Reportergen-Plasmides zur Expression des Transgens *in situ* führte, ist die DNA-Vakzinierung zu einer vielversprechenden Alternative zu Protein- und Peptidimpfstoffen geworden. Dafür sprechen zum einen biologische Gründe wie die Generierung starker T_h1- und CTL-Antworten durch die endogene und lang andauernde Expression des Antigens, welches dem Immunsystem nativ gefaltet und mit korrekten posttranslatorischen Modifikationen versehen präsentiert werden kann [Liu 2000].

Andererseits bieten DNA-Vakzine auch praktische Vorteile wie kostengünstige Herstellung, hohe Stabilität und damit einfache Lagerung bzw. Transport.

Leider wird das Antigen auch aus den unter „Steigerung der Expression" angeführten Gründen bei der Verwendung von natürlichen Genen oft nur in geringen Mengen exprimiert. Durch die Verwendung Kodonoptimierter Gene kann zum einen die Translation verbessert und gleichzeitig im natürlichen Gen vorhandene expressions-limitierende cis-aktive Sequenzmotive ausgeschaltet werden. Durch die Einführung immunstimulierender CpG-Motive in die Sequenz des synthetischen Gens oder auch des Vektor-Backbones können DNA-Impfstoffe quasi mit einem „eingebauten" Adjuvans ausgestattet werden, welches die Immunantwort deutlich verstärken kann. Einen unter Sicherheitaspekten positiven „Nebeneffekt" der Kodonoptimierung stellt die verringerte Sequenzhomologie zu wildtyp-Sequenzen dar. Dadurch können Rekombinationsereignisse beispielsweise mit wildtyp-Pathogenen, welche u.U. zu Chimären mit neuartigen pathogenen Eigenschaften führen könnten, sicher unterbunden werden. Durch die Tatsache, dass synthetische Gene aus künstlichen Oligonukleotiden aufgebaut werden, kann auch die Verschleppung potentiell hochpathogenen Materials (beispielsweise replikationsfähigen Viren) bei der Isolierung von DNA aus S3- und S4-Organismen ausgeschlossen werden. Idealerweise wird durch „Database-Cloning" der Umgang mit letzteren in vielen Anwendungsfällen vollständig vermieden.

Die Brücke zur Kombinatorik schlägt die Technik des „Gene scrambling". Dabei wird die Aminosäuresequenz eines Proteins in kürzere Segmente unterteilt, welche anschließend in anderer Reihenfolge wieder zusammengefügt werden. Idealerweise wird dabei die Untergliederung bzw. die Neuanordnung so vorgenommen, dass sich die Enden der Segmente um einige möglichst ähnliche oder identische Aminosäuren überlappen. Durch diese Technik kann die ursprüngliche Funktionsweise des Proteins eliminiert werden, ohne aber gleichzeitig für die Immunerkennung wichtige Epitope zu zerstören. Die so erhaltene Aminosäuresequenz wird dann unter Berücksichtigung der unter 1.1.2 angeführten Kriterien rückübersetzt und kann in einer DNA-Vakzine Verwendung finden [Thomson 2001].

A.4 Möglichkeiten und Limitationen bestehender Gensynthese-Software

Da im genetischen Code mit Ausnahme von Methionin und Tryptophan jede Aminosäure durch mehrere verschiedene Kodons repräsentiert wird, eröffnet sich beim Design synthetischer Gene die Möglichkeit, die DNA-Sequenz an die experimentellen Erfordernisse anzupassen, ohne dabei die Aminosäuresequenz des codierten Proteins zu verändern.

Eines der ältesten Werkzeuge zum Design synthetischer Gene, allerdings immer noch häufig benutzt, sind Rückübersetzungsprogramme („Backtranslation-tools"), welche eine gegebene Aminosäuresequenz unter Berücksichtigung der für den Zielorganismus optimalen Kodonwahl in die entsprechende DNA-Sequenz übersetzen. Berücksichtigt man den enormen Zeit- und Kostenaufwand, den die Herstellung eines synthetischen Gens in der Frühzeit der Gensysnthese erforderte, so wird verständlich, dass oftmals nicht nur die optimale Kodonwahl, sondern vielmehr auch die universelle Verwendbarkeit des Konstruktes angestrebt wurde. Eine elegante Möglichkeit, ein synthetisches Gen im nachhinein zu verändern, stellt die Kassetten-Mutagenese dar, d.h. man betrachtet das Gen als Kette kurzer DNA-Blöcke, deren Ränder durch in der Gesamtsequenz einzigartige Restriktions-Schnittstellen definiert sind. Dadurch lässt sich später leicht ein Block mittels Restriktionsverdau/Ligation durch einen anderen ersetzen. Folgerichtig wurden mehrere Programme entwickelt, die, zumeist ausgehend von einer degenerierten DNA-Sequenz (d.h. die variablen Basen werden durch die entsprecheden IUPAC-Symbole dargestellt), die Lokalisation potentieller Schnittstellen und ggf. deren Einführung in die Sequenz des synthetischen Gens gestatten. Idealerweise werden für die nicht durch Schnittstellen festgelegten Sequenzabschnitte auf den Zielorganismus abgestimmte Kodons verwendet [Presnell 1988, Makarova 1992, Raghava 1994, Libertini 1992, Weiner 1989].

Umfangreichere Programme berücksichtigen auch die mögliche Ausbildung von Repetitionen oder Haarnadelschleifen in der Sequenz, welche die Synthese und unter Umständen die Expression des Gens beeinträchtigen könnten.

Petri et al. präsentieren ein Programm, welches eine recht spezielle Form der Gensynthese unterstützt. Es bietet jedoch neben Kodonoptimierung auch die Möglichkeit, automatisch Subfragmente für die Gensynthese zu entwerfen, entsprechende Schnittstellen einzuführen und Haarnadelschleifen zu eliminieren [Petri 1989].

Das Programmpaket *Synsos* bietet neben einem Sequenzeditor und der obligatorischen Optimierung auf häufig genutzte Kodons auch die Möglichkeit, eine Restriktionsanalyse (allerding ohne automatische Einführung von Schnittstellen) und eine Überprüfung der Sequenz auf mögliche (invertierte) Repetitionen durchzuführen. Darüber hinaus wird ein Algorithmus implementiert, der geeignet ist, größere wiederhohlte oder invertiert wiederholte DNA-Abschnitte zu eliminieren [Ochagavia 1992].

Die neueste Veröffentlichung im Bereich softwaregestützte Gensynthese stellt das Programm *DNAWorks* vor, welches den Genaufbau via PCR unterstützt. Nach einer Kodonoptimierung wird eine grobe Aufsplittung der Sequenz in eine gerade Zahl von Bereichen mit ähnlicher Schmelztemperatur vorgenommen. Anschließend wird eine

Variante des *Simulated Annealing* Algorithmus benutzt, um durch Kodonvariation und leichte Veränderung der Bereichsgrenzen letztlich Oligonucleotide mit geringer Tendenz zur Haarnadelbildung und homogener Schmelztemperatur der Überlappungen zu erhalten [Hoover 2002].
Ohne Veränderung der initial vorgegebenen Sequenz versucht das Programm *Regen* auszukommen, welches den Syntheseprozess einer ligationsbasierten Eintopf-Gensynthese unterstützt. Hier hybridisieren bei hohen Temperaturen jeweils zwei Oligos dergestalt, dass nur noch kurze einzelsträngige Enden erhalten bleiben. Bei niedrigeren Temperaturen können diese bereits größtenteils doppelsträngigen Fragmente mittels der kurzen Überhänge zu einem kompletten Gen ligiert werden. Die Software wählt die Oligogrenzen nun so, dass beim Ligieren Fehlhybridisierungen nicht zusammengehöriger Fragmente nahezu ausgeschlossen sind [Jerala 1988].

Alle vorgestellten Programme sind jedoch in ihrer Funktionalität auf Teilaspekte der Gensynthese limitiert, d.h. keines integriert sämtliche Aspekte der Gensynthese (Optimierung, Analyse und Produktionsunterstützung) in einem homogenen Paket. Vor allem die vielfältigen Chancen des rationalen Gendesigns werden nur unzureichend genutzt, so bietet z.B. kein Programm die Möglichkeit, den GC-Gehalt anzupassen. Darüber hinaus sind die meisten der Softwarepakete nicht für den Betrieb unter einem modernen graphischen Betriebssytem geeignet.

A.5 Zielsetzung und Rahmenbedingungen

Ziel der vorliegenden Arbeit war, sämtliche in einer Hochdurchsatz-Produktionsumgebung für Gensynthese anfallenden Arbeitsschritte - soweit möglich - durch geeignete Software zu unterstützen. Dabei muss eine größtmögliche Flexibilität hinsichtlich des Sequenzdesigns (es sollen synthetische Gene variabler Länge und für unterschiedlichste Anforderungen synthetisiert werden) und eine hohe Stabilität des Produktionsprozesses angestrebt werden. Prinzipiell lassen sich die Einsatzgebiete in drei Bereiche gliedern:

A.5.1 Design und Optimierung

Aufgrund der *de-novo*-Synthese können synthetische Gene quasi „am Reißbrett" entworfen werden. Dieses soll die Software in Form eines auf die Anwendung „Gendesign" hin ausgerichteten und mit allen Standardfunktionalitäten ausgestatteten Sequenzeditors zur Verfügung stellen. Neben manuellen Editiermöglichkeiten soll aber vor allem auch eine vollautomatische Sequenzoptimierung zur Verfügung stehen. Die Software soll dazu unter Ausnutzung

der Degeneriertheit des genetischen Codes eine Sequenz ermitteln, die den vorgegebenen experimentellen Anforderungen hinsichtlich Kodonwahl, GC-Gehalt, DNA-Motive etc. möglichst nahe kommt.

A.5.2 Sequenzanalyse

Komplementär zu den Editier- und Optimierungsmöglichkeiten sind Funktionen zur Sequenzanalyse und Annotation. Konsequenterweise muss eine vorliegende DNA-Sequenz auf jeden variablen Sequenzparameter hin analysiert werden können. Die Ergebnisse sollen dem Anwender in Form anschaulicher Graphen (z.B. GC-Gehalt) oder Berichte (z.B. Sequenzannotation) präsentiert werden. Idealerweise resultiert aus der Kombination von Optimierung und Analyse ein Feedback-Prozess, bei dem der Anwender z.B. nach einer automatischen Optimierung die Sequenz analysiert und daraufhin die Optimierung mit leicht anderer Gewichtung der Optimierungsparameter neu startet, um sich so iterativ dem bestmöglichen Kompromiss zwischen u.U. konträren Zielvorstellungen in Bezug auf unterschiedliche Kriterien anzunähern.

A.5.3 Produktionsunterstützung

Steht die zu synthetisierende Sequenz fest, muss sie bei längeren Genen in Subfragmente untergliedert werden. Dieser Prozess sowie das Anfügen entsprechender Linker an die Fragmentenden soll durch die Software unter Beachtung vorgegebener Rahmenbedingungen, wie. z.B. maximale Fragmentlänge, unterstützt werden. Bei der anschließenden Generierung der benötigten Oligonukleotide muss die Software die Länge der Hybridisierungsstrecken zueinander komplementärer Oligonucleotidabschnitte so wählen, dass eine vorgegebene Schmelztemperatur möglichst genau eingehalten wird, um einen stabilen und zuverlässigen Batch-Ligationsprozess zu ermöglichen. Darüber hinaus soll der Anwender vor potentiellen Produktionsproblemen, z.B. aufgrund möglicher Fehlhybridisierungen, gewarnt werden.
Wie bereits in Abschnitt A.2 beschrieben, müssen aufgrund der unabdingbar bei der Oligosynthese auftretenden Mutationen zahlreiche Subfragment-Klone auf die richtige Sequenz hin durchgemustert werden. Die vom Sequenzer gelieferten Sequenzdateien sollen daher durch die Software soweit analysiert und in Hinblick auf Fehlerfreiheit bewertet werden, dass der Anwender idealerweise nur noch das als am besten bewertete Chromatogramm zur Bestätigung visuell kontrollieren muss.

A.5.4 Rahmenbedingungen

Soweit sinnvoll, sollten alle Schritte in einer integrierten Arbeitsumgebung durchgeführt werden können. Da in der Produktionsumgebung bereits durchgängig Windows 2000 eingesetzt wurde, sollte die Software auch unter diesem Betriebssystem laufen und sich dabei an den Bedienstandards dieser grafischen Benutzerschnittstelle orientieren. Zur Datenspeicherung- und verwaltung bot sich die Verwendung eines relationalen Datenbanksystems an. In Hinblick auf die spätere Verknüpfung mit einem Standard-Laborinformationsmanagementsystem bzw. Robotersystemen zur Automatisierung fiel die Wahl auf Oracle.

B Materialien & Methoden

B.1 Grundlegende Techniken der Bioinformatik

Während der Entwicklung der GeneOptimizer-Suite wurde vor allem auf zwei grundlegende Methodenbereiche der Bioinformatik zurückgegriffen. Dies sind zum einen Alignment-Algorithmen zur Evaluierung der Ähnlichkeit zweier Sequenzen. Ebenso wichtig sind Techniken zur Repräsentation von DNA-Sequenzmotiven, welche z.B. DNA-Proteininteraktionsstellen charakterisieren können, und zum Auffinden und Bewerten derselben innerhalb einer vorgegebenen Sequenz. Nachfolgend sollen diese Techniken einführend vorgestellt werden.

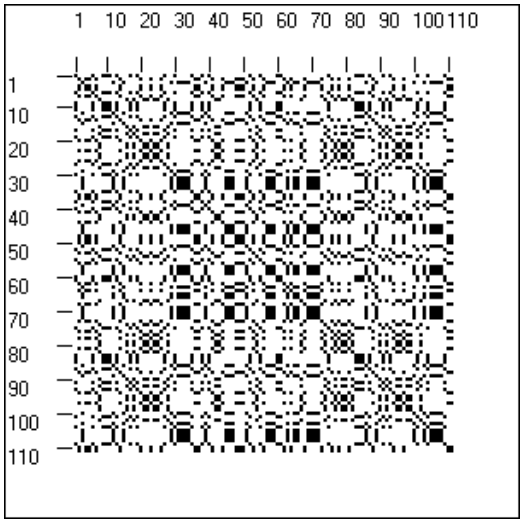
B.1.1 Alignment-Algorithmen

B.1.1.1 Einführung

Die fundamentale Bedeutung von Algorithmen zum Sequenzvergleich liegt vor allem darin begründet, dass eine hohe Sequenzähnlichkeit in der Primärstruktur von Proteinen und DNA oft auch eine funktionelle oder strukturelle Ähnlichkeit impliziert. So können beim Vergleich zweier Proteinsequenzen hochkonservierte und damit funktionell bedeutende Domänen aufgespürt werden, oder Rückschlüsse auf die Funktion eines Gens durch Sequenzvergleich der DNA mit dem Genom eines gut erforschten Modellorganismus gezogen werden. Die dafür entwickelten Algorithmen können jedoch auch dazu benutzt werden, in der Sequenz liegende Besonderheiten, wie Wiederholungen ähnlicher DNA-Motive, aufzuspüren.

Eine einfache Möglichkeit, Sequenzähnlichkeiten aufzufinden, bietet die Technik des „Dot-Plot". Dabei definieren die beiden Sequenzen die Achsen eines Koordinatensystems, in welches eine Markierung eingetragen wird, sofern die entsprechende Koordinate durch einander entsprechende Basen definiert wird.

Stark homologe Sequenzbereiche lassen sich dabei intuitiv an den über längere Strecken durchgezogenen Diagonalen erkennen. Um einen computergestützten Vergleich zweier Zeichenketten durchzuführen, muss jedoch der Begriff der Ähnlichkeit zunächst stärker formalisiert werden [Gusfield 1999].

a; 

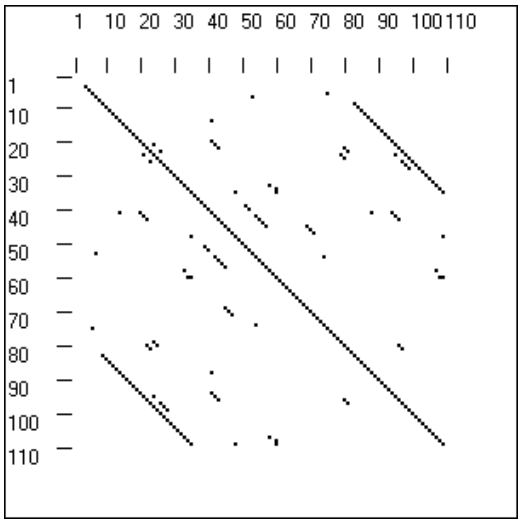
b; 

Abb. B.1.1.1-1 Dot-plot einer Sequenz mit sich selbst, deren Bereiche 7-37 und 81-111 bis auf eine Base identisch sind. Diese Repetition erkennt man bereits im Dotplot a als Diagonalen in der unteren linken und oberen rechten Ecke. Durch Anwendung eines Filters, der ein 6-Basen langes Fenster über alle Diagonalen der Matrix schiebt und in der Mitte des Fensters einen Punkt setzt, sobald im Originalplot wenigstens 5 Entsprechungen vorhanden sind, werden starke Repetitionen leichter erkennbar.

Def.: Man erhält ein (globales) Alignment zweier Zeichenketten S_1 und S_2, indem fakultativ zunächst definiert Leerzeichen entweder in die Zeichenketten eingefügt oder an die Enden der Zeichenketten S_1 und S_2 angefügt werden und die beiden resultierenden Zeichenketten dann übereinander gestellt werden, so dass jedes Zeichen oder jede Leerstelle in der einen Zeichenkette einem einzigen Zeichen oder einer Leerstelle in der anderen Zeichenkette gegenübergestellt ist.

```
a; AGCC-GTGT        b; AGCC-GTGT
   :|||:|:||           ::|::::::
   -GCCTGAGT           GCCTGAGT-
```

Abb. B.1.1.1-2 Zwei Alignments der Sequenzen „AGCCGTGT" und „GCCTGAGT"

So stellt sowohl Abb. B.1.1.1-2a als auch Abb. B.1.1.1-2b ein mögliches Alignment der Sequenzen „AGCCGTGT" und „GCCTGAGT" dar, wobei offensichtlich ist, dass das Alignment a die Ähnlichkeit der Sequenzen besser verdeutlicht. Wir müssen also einen Weg finden, diese höhere „Qualität" des Alignments a durch einen numerischen Wert auszudrücken und einen Algorithmus entwickeln, der es uns gestattet, das optimale Alignment der beiden Sequenzen zu finden.

Def.: Sei Λ das für die Zeichenketten S_1 und S_2 benützte Alphabet und Λ' das Alphabet Λ mit dem zusätzlichen Zeichen „-„ welches eine Leerstelle symbolisiert.

Dann bezeichnet *s(x,y)* für zwei beliebige Zeichen *x,y* aus Λ den Wert (den „Score"), der durch das Alignment des Zeichens *x* mit dem Zeichen *y* erhalten wird.

	A	T	C	G	-
A	5	-1	-1	-1	-2
T		5	-1	-1	-2
C			5	-1	-2
G				5	-2
-					0

Abb. B.1.1.1-3 Scoring-Matrix für DNA. Die Identität zweier Basen wird mit 5 Punkten bewertet, die Nichtidentität mit –1. Das Einfügen einer Leerstelle „-„ wird mit –2 gewichtet.

Die paarweisen Scores *s(x,y)* werden zweckmäßigerweise mit Hilfe einer Scoring-Matrix dargestellt. Abb. B.1.1.1-3 zeigt eine solche Matrix für DNA. Üblicherweise wird in Alignment-Problemen *s(x,y)* größer 0 gesetzt, falls sich die Zeichen *x* und *y* entsprechen, andernfalls kleiner 0 bzw. *s(x,y)* ist umso größer, je „ähnlicher" sich *x* und *y* sind. Große Bedeutung haben Scoring-Matrices vor allem beim Vergleich zweier Proteinsequenzen. Aminosäuren haben aufgrund ihrer biochemischen und physikalischen Eigenschaften eine unterschiedliche Tendenz, im Laufe der Evolution gegeneinander ausgetauscht zu werden, ohne dass die biologische Funktion des Proteins beeinträchtigt wird. So ist es z.B. sinnvoll, dem Alignment zweier Aminosäuren einen umso höheren Score zuzuordnen, je ähnlicher sie sich in ihren Eigenschaften (z.B. Hydophobizität) sind.

Def.: Für ein gegebenes Alignment *A* von S_1 und S_2, seien S_1' und S_2' die Zeichenketten nach dem definierten Einfügen von Leerstellen und *l* die Länge der gleichlangen Zeichenketten S_1' und S_2' in *A*. Der Wert (der Score) von *A* ist definiert als $\sum_{i=1}^{l} s\left(S_1'(i), S_2'(i)\right)$. Für eine gegebene paarweise Scoringmatrix für das Alphabet Λ' ist die Ähnlichkeit zweier Zeichenketten S_1 und S_2 definiert als der Score des Alignments *A* von S_1 und S_2, welches den Alignmentscore maximiert. Dies ist der optimale Alignment-Score für die Zeichenketten S_1 und S_2.

Def.: Sei *V*(*i,j*) der optimale Alignmentscore der Prefixe $S_1[1..i]$ und $S_2[1..j]$.
V(*i,j*) ist also der Alignmentscore, den man erhält, wenn man die ersten *i* Buchstaben von S_1 mit den ersten *j* Buchstaben von S_2 optimal aligned. Besteht S_1 aus *n* Zeichen und S_2 aus *m* Zeichen, so bezeichnet *V*(*n,m*) den optimalen Alignment-Score der Zeichenketten S_1 und S_2.
(Alle Definitionen übersetzt aus [Gusfield 1999])

Die Basisbedingungen

F. B.1.1.1-1,2

$$V(0,j) = \sum_{1 \le k \le j} s\left(-, s_2(k)\right)$$

$$V(i,0) = \sum_{1 \le k \le i} s\left(s_1(k), -\right)$$

stellen den Wert des Alignments der ersten *j* bzw. *i* Zeichen der Zeichenketten S_1 bzw. S_2 mit Leerstellen dar. Zwischen dem Wert für $V(i,j)$ und den Werten für *V* mit Indexpaaren kleiner als *i* und *j* lässt sich folgende rekursive Beziehung aufstellen:

F. B.1.1.1-3

$$V(i,j) = \max \begin{pmatrix} V(i-1,j-1) + s(S_1(i), S_2(j)) \\ V(i-1,j) + s(S_1(i), -) \\ V(i,j-1) + s(-, S_2(j)) \end{pmatrix}$$

Auf einen exakten Beweis soll an dieser Stelle verzichtet werden, jedoch wird die Bedeutung dieser Rekursion weiter unter anschaulich klar werden.

B.1.1.2 Berechnung des Alignmentscores

Prinzipiell ließe sich die o.g. Rekursion mit fast jeder Programmiersprache leicht implementieren. Jedoch steigt die Zahl der Aufrufe einer Funktion zur Berechnung von $V(i,j)$ mit wachsendem *m* und *n* exponentiell an. Hierbei werden massiv redundante Aufrufe der rekursiven Funktion vorgenommen, die sich allerdings bei einer tabellarischen Berechnung von $V(i,j)$, dem sogenannten „Dynamic Programming" vermeiden lassen. Dazu wird eine Tabelle der Größe $(n+1)*(m+1)$ benötigt.

Die Werte der Spalte 0 und der Zeile 0 ergeben sich direkt aus den Basisbedingungen. Anschließend kann die Tabelle z.B. spaltenweise mit ansteigendem *i* und *j* anhand der rekursiven Beziehung für $V(i,j)$ aufgefüllt werden. Anschaulich evaluiert diese Beziehung also, auf welchem Wege man am besten von den links, oben und diagonal angrenzenden Zellen zur Zelle $V(i,j)$ kommt und ob die korrespondierenden Zeichen $S_1(i)$ und $S_2(j)$ aligned werden oder statt dessen im finalen Alignment eine Leerstelle in S_1 bzw. S_2 eingefügt wird.

Eine Analogie der Dynamic-Programming-Tabelle zum Dotplot ist unverkennbar. Nach vollständiger Berechnung der DP-Tabelle kann der optimale Alignment-Score direkt aus Zelle (n,m) entnommen werden.

			A	G	C	C	G	T	G	T
		0	1	2	3	4	5	6	7	8
	0	0	-1_l	-2_l	-3_l	-4_l	-5_l	-6_l	-7_l	-8_l
G	1	-1_u	-1_d	0_d	-1_l	-2_l	-3_d	-4_l	-5_d	-6_l
C	2	-2_u	-2_d	-1_u	1_d	0_d	-1_l	-2_l	-3_l	-4_l
C	3	-3_u	-3_d	-2_u	0_d	2_d	1_l	0_l	-1_l	-2_l
T	4	-4_u	-4_d	-3_u	-1_u	1_u	1_d	2_d	1_l	0_d
G	5	-5_u	-5_d	-3_d	-2_u	0_u	2_d	1_u	3_d	2_l
A	6	-6_u	-4_d	-4_u	-3_u	-1_u	1_u	1_d	2_u	2_d
G	7	-7_u	-5_u	-3_d	-4_u	-2_u	0_d	0_d	2_d	1_d
T	8	-8_u	-6_u	-4_u	-4_d	-3_u	-1_u	1_d	1_u	3_d

Abb. B.1.1.2-1 Vollständig gefüllte Dynamic-Programming-Tabelle. Die Zelleninhalte ergeben sich mit den Parametern 1 für Matches und –1 für Mismatches und Gaps. Die Buchstaben stellen die während der Berechnung hinterlassenen Zeiger u für up, l für left und d für diagonal dar (Es ist pro Zelle nur ein Zeiger gezeigt)

B.1.1.3 Generierung des Alignments

Wenngleich man nun einen die Ähnlichkeit von S_1 und S_2 charakterisierenden Wert erhalten hat, ist es oftmals wünschenswert, auch das vollständige Alignment zu generieren. Dazu hinterlässt man idealerweise bereits während der Berechnung des DPT entsprechende Zeiger in der Tabelle. Konkret heißt das bei der Berechnung von $V(i,j)$, setze einen Zeiger von Zelle (i,j) nach Zelle $(i\text{-}1,j\text{-}1)$ falls $V(i,j) = V(i-1,j-1) + s(S_1(i), S_2(j))$, falls $V(i,j) = V(i-1,j) + s(S_1(i),-)$ nach$(i\text{-}1,j)$ und nach Zelle $(i,j\text{-}1)$ falls gilt $V(i,j) = V(i,j-1) + s(-,S_2(j))$. Dabei kann es durchaus vorkommen, dass ausgehend von Zelle (i,j) mehrere Zeiger in unterschiedliche angrenzende Zellen weisen.

Ein optimales Alignment wird nun erhalten, indem man einem möglichen durch die Zeiger vorgegebenen Pfad von Zelle (n,m) zu Zelle (0,0) folgt. Dabei interpretiert man eine Diagonale (von Zelle (i,j) zu $(i\text{-}1,j\text{-}1)$) als „Match", falls $S_1(i)=S_2(j)$, andernfalls als Substitution. Ein Zeiger in die obere Zelle ((i,j) nach $(i\text{-}1,j)$) stellt im Alignment eine Insertion des Zeichens $S_1(i)$ dar, entsprechend wird für das Alignment nach $S_2(j)$ eine Leerstelle in S_2 eingefügt. Analog wird ein Zeiger in die linke Zelle $(i,j\text{-}1)$ als Insertion in S_2 berücksichtigt. (Wobei Insertion in S_1 natürlich auch als Deletion in S_2 gesehen werden kann und *vice versa*.)

			A	G	C	C	G	T	G	T
		0	1	2	3	4	5	6	7	8
	0	0	-1_l	-2_l	-3_l	-4_l	-5_l	-6_l	-7_l	-8_l
G	1	-1_u	-1_d	0_d	-1_l	-2_l	-3_d	-4_l	-5_d	-6_l
C	2	-2_u	-2_d	-1_u	1_d	0_d	-1_l	-2_l	-3_l	-4_l
C	3	-3_u	-3_d	-2_u	0_d	2_d	1_l	0_l	-1_l	-2_l
T	4	-4_u	-4_d	-3_u	-1_u	1_u	1_d	2_d	1_l	0_d
G	5	-5_u	-5_d	-3_d	-2_u	0_u	2_d	1_u	3_d	2_l
A	6	-6_u	-4_d	-4_u	-3_u	-1_u	1_u	1_d	2_u	2_d
G	7	-7_u	-5_u	-3_d	-4_u	-2_u	0_d	0_d	2_d	1_d
T	8	-8_u	-6_u	-4_u	-4_d	-3_u	-1_u	1_d	1_u	3_d

Abb. B.1.1.3-1 Das globale Alignment wird erhalten, indem man den Zeigern u, l und d von Zelle (8,8) bis Zelle (0,0) folgt.

Da von einer Zelle mehrere Zeiger ausgehen können, gibt es u.U. auch mehrere in ihrem Score äquivalente Alignments.

Ein so erhaltenes Alignment wird als globales Alignment oder nach den Entwicklern des Algorithmus auch als „Needleman-Wunsch"-Alignment bezeichnet.

Ein Spezialfall des globalen Alignments ergibt sich, wenn *a priori* bekannt ist, dass eine Zeichenkette innerhalb der anderen enthalten ist oder sich die Enden der Zeichenketten überlappen. Um in diesen Fällen ein korrektes Alignment zu erhalten, dürfen die im Alignment am Anfang oder am Ende einer Zeichenkette eingefügten Leerstellen nicht mit einem negativen Score belegt werden, sondern werden mit 0 gewichtet, um ein sogenantes „*end-space-free*"-Alignment zu erhalten.

Dieses wird z.B. bei der „Shotgun-Sequenzierung", aus der man eine große Anzahl von teilweise einander überlappenden Sequenzfragmenten erhält, angewendet. Indem für alle möglichen Paare der Sequenzfragmente ein end-space free Alignment durchgeführt wird, kann durch überlappendes Aneinanderfügen der Paare, welche einen möglichst hohen Alignmentscore erzielen, die Gesamtsequenz rekonstruiert werden.

Zur Implementierung des „*end-space-free*"-Alignments kann die Rekursions-bedingung des globalen Alignments verwendet werden. Um an den Anfang einer Zeichenkette gesetzte Leerzeichen zu vernachlässigen, müssen jedoch die Basisbedingungen angepasst werden:

F. B.1.1.3-1,2

$$V(0,j) = 0 \quad 0 \le j \le n$$

$$V(i,0) = 0 \quad 0 \le i \le m$$

Der Score für das optimale Alignment findet sich jedoch nicht notwendigerweise in Zelle (*n*,*m*), sodern ist vielmehr der Maximalwert über alle Zellen in Zeile *n* oder Spalte *m*.
Damit wird der gewünschten Vernachlässigung der im Alignment ans Ende einer Zeichenkette angefügten Leerstellen Rechnung getragen. So entsprechen z.B. die Zellen in Zeile *n* Alignments, bei denen das letzte Zeichen von Zeichenkette S_1 zum Score beiträgt, aber rechtsgelegene Zeichen von S_2 nicht.

B.1.1.4 Lokale Alignments

Globale Alignments eignen sich besonders, um Sequenzen zu vergleichen, die über ihre ganze Länge eine hohe Ähnlichkeit zueinander aufweisen, wie z.B. verschiedene Proteine einer Proteinfamilie. Um jedoch z.B. hochkonservierte funktionelle Untereinheiten von in ihrer Gesamtheit sehr unterschiedlichen Proteinen (wie z.B. bei unterschiedlichen Proteinfamilien) aufzuspüren, wird ein Verfahren benötigt, welches in längeren Sequenzen zueinander ähnliche Bereiche finden kann. Es dauerte mehr als zehn Jahre nach Entwicklung des globalen Alignments, bis Smith und Waterman einen „Local Alignment"-Algorithmus vorstellten. Dabei handelt es sich im wesentlichen um das bereits beschriebene Global-Alignment-Verfahren, jedoch mit „end-space free" – Basisbedingungen und einer erweiterten rekursiven Beziehung. Diese ergeben sich anschaulich aus der Tatsache, dass ein lokales Alignment an beliebiger Stelle innerhalb zweier Sequenzen beginnen kann, ungeachtet der Anzahl führender Zeichen. Der Wert für $V(i,j)$ darf also nie unter 0 absinken.

F. B.1.1.4-1,2

$$V(0,j) = 0$$
$$V(i,0) = 0$$

F. B.1.1.4-3

$$V(i,j) = \max \begin{pmatrix} 0 \\ V(i-1,j-1) + s(S_1(i), S_2(j)) \\ V(i-1,j) + s(S_1(i), -) \\ V(i,j-1) + s(-, S_2(j)) \end{pmatrix}$$

Den Score für das optimale lokale Alignment findet man nun in der Zelle mit dem höchsten Wert der gesamten Tabelle. Um das lokale Alignment zu generieren, folgt man ausgehend von dieser Zelle den wie üblich bei der Berechnung der DPT gesetzten Zeigern bis ein Eintrag mit dem Wert 0 erreicht wird. Oftmals soll jedoch nicht nur ein optimales lokales Alignment, sondern alle, ggf. auch suboptimalen Alignments generiert werden, deren Score über einem bestimmten Grenzwert liegt und die sich nicht überlappen. Dazu führten Waterman und Eggert einen Algorithmus

ein, der es gestattet, ein suboptimales Alignment zu erhalten ohne dass dazu vorher ein Großteil der DPT neu berechnet werden muss [Waterman 1997].
Beginnend mit der linken oberen Zelle (*i*, *j*) des zuletzt erhaltenen Alignments wird folgende Formel angewendet, um die veränderten Matrixelemente $V^*(i, j)$ zu berechnen:

F. B.1.1.4-4

$$V^*(i,j) = \max\begin{pmatrix} 0 \\ V(i-1,j) + s(S_1(i),-) \\ V(i,j-1) + s(-,S_2(j)) \end{pmatrix}$$

Die Neuberechnung wird für die Spalte *j* für alle Zellen (*k,j*) mit *k*>*i* durchgeführt, bis der neue Wert $V^*(k,j)$ identisch mit dem alten Wert $V(k,j)$ ist. Analog werden die Matrixelemente für die Zellen (*i,k*) mit *k*>*j* in der Zeile *i* neu berechnet, bis wiederum gilt $V^*(i,k) = V(i,k)$. Diese Neuberechnungen werden ausgehend von allen (*i,j*) des vorhergehenden Alignments durchgeführt, wobei mindestens bis zu der Spalten- bzw. Zeilenposition neu berechnet werden muss, wie es in der vorhergehenden Zeile oder Spalte notwendig war. Nach Abschluss der Neuberechnungen kann das suboptimale Alignment wie gewohnt erhalten werden.

Lokale Alignments können nicht nur zum Auffinden von Sequenzhomologien zwischen zwei unterschiedlichen Sequenzen verwendet werden, sondern auch, um in einer Sequenz einander ähnliche Bereiche (Repetitionen) zu identifizieren [Waterman 2000]. Dazu wird ein lokales Alignment der Sequenz mit sich selbst durchgeführt. Um zu verhindern, dass als optimales Alignment das Alignment der vollständigen Sequenz mit sich selbst generiert wird, muss die Diagonale mit $V(i,i) = 0$ initialisiert werden. Aus Symmetriegründen ist es ausreichend, lediglich $V(i,j)$ für *i*<*j* zu berechnen.

B.1.2 Motivdarstellung und Suche

Lokale Erkennungssequenzen, z.B. für Protein-DNA-Interaktionen, spielen in der gesamten Zell- und Molekularbiologie eine entscheidene Rolle. Daher gehört das Auffinden entsprechender Motive, z.B. in der Genomannotation, zu den wichtigsten Aufgaben der Bioinformatik.
Leider sind nur wenige Erkennungsmotive, wie z.B. Restriktionsenzym-Schnittstellen, als eindeutige DNA-Sequenzen definiert. Statt dessen findet man bei Betrachtung des Alignments funktionell identischer Bindungsstellen, dass an manchen Positionen z.B. zwei unterschiedliche Basen auftreten können.

```
Position:  1 2 3 4 5 6
Seq. 1:    A C T T G T
Seq. 2:    T C T T G T
Seq. 3:    A C T T C T
Seq. 4:    T C T T G A
Seq. 5:    C C T T G A
Seq. 6:    C C T T C T
Consensus: H C T T S W
```

Abb. B.1.2-1 Alignment sechs leicht unterschiedlicher, aber funktionell identischer Bindungsstellen.

Durch ein erweitertes Alphabet (IUPAC-Nomenklatur), in welchem ein Buchstabe mehrere verschiedene Basen codieren kann, lässt sich aus dem Alignment eine Consensus-Sequenz ableiten.

A	A	Y	C,T
C	C	H	A,C,T
G	G	R	A,G
T	T	D	A,G,T
S	C,G	B	C,G,T
K	G,T	N	A,C,G,T
V	A,C,G	W	T,A
M	A,C		

Abb. B.1.2-2 Das erweiterte Alphabet nach IUPAC-Nomenklatur

Dieses Verfahren weist jedoch eine Reihe von Nachteilen auf. Zum einen müssen die Consensus-Zeichen nach einer willkürlichen Mehrheitsregel festgelegt werden. Diese kann z.B. besagen, dass ein eindeutiges Basensymbol (A,C,G oder T) verwendet

wird, wenn in mehr als 80 Prozent der Sequenzen an einer Position eine bestimmte Base auftritt. Auch lässt die Verwendung eines Mehrdeutigkeitssymboles wie z.B. Y keine Aussage über das Häufigkeitsverhältnis der Basen C und T zu. Weiterhin erhält man bei der Analyse einer unbekannten Sequenz unter Verwendung der IUPAC-Consensi in der Regel nur eine Ja/Nein Aussage, die Gefahr, eine real vorhandene Bindungsstelle zu übersehen ist sehr groß. Wird jedoch eine dem Consensus entsprechende Sequenz aufgefunden, lässt sich die „Qualität" (z.B. hinsichtlich Erkennungssicherheit oder biologischer Aktivität) der Bindungsstelle nicht quantifizieren.
Diese Nachteile können weitgehend vermieden werden, indem aus dem Alignment bekannter Bindungssequenzen eine Matrixdarstellung generiert wird. Diese erhält man im einfachsten Fall, indem für jede Position die Anzahl der vier Nucleotide aufnotiert wird.

Position *i*:	**1**	**2**	**3**	**4**	**5**	**6**
Seq. 1:	A	C	T	T	G	T
Seq. 2:	T	C	T	T	G	T
Seq. 3:	A	C	T	T	C	T
Seq. 4:	T	C	T	T	G	A
Seq. 5:	C	C	T	T	G	A
Seq. 6:	C	C	T	T	C	T
Consensus:	**H**	**C**	**T**	**T**	**S**	**W**
Σ_i A	2	0	0	0	0	3
Σ_i T	2	0	6	6	0	3
Σ_i G	0	0	0	0	4	0
Σ_i C	2	6	0	0	2	0

Abb. B.1.2-3 Berechnung der Häufigkeitsmatrix

Eine Testsequenz kann nun dergestalt analysiert werden, dass die Matrix über die gesamte Sequenz im Abstand eines Nucleotides verschoben wird und für jede Position ein Score, der die Ähnlichkeit des Testsequenzabschnitts mit der Matrix charakterisiert, errechnet wird.
Es ist einleuchtend, dass der Informationsgehalt einer Position, an welcher ausschließlich eine Base auftritt, wesentlich höher ist, als wenn drei unterschiedliche Basen relativ gleichverteilt auftreten können. Es ist also sinnvoll, eine weitere Gewichtung der Matrixpositionen einzuführen [Quandt 1995]. Dazu wird für jede Spalte *i* ein Consensus Index (C_i) errechnet, welcher wie folgt definiert wird.

F. B.1.2-1

$$C_i = 100/\ln 4 * \left(\sum_{b=A,C,G,T} p_{i,b} * \ln p_{i,b} + \ln 4 \right)$$

C_i erreicht also den höchsten Wert 100, wenn an einer Position ausschließlich ein Nucleotid angetroffen wird, und wird 0 für ein gleichverteiltes Auftreten aller vier Basen. Zur Bewertung eines der Motivlänge n entsprechenden Abschnittes der Testsequenz wird für diesen ein Matrixähnlichkeitswert *mat_sim* gemäß F. B.1.2-2 errechnet, wobei $score_{i,b}$ den Matrixwert für Base *b* an Position *i* darstellt und $max\_score_{i,b}$ = max($score_{i,b}$) mit b ∈ A,G,C,T. Dies ist beispielhaft in Abb. B.1.2-4 illustriert.

F. B.1.2-2

$$mat\_sim = \frac{\sum_{i=1}^{n} C_i * score_{i,b}}{\sum_{i=1}^{n} C_i * max\_score_{i,b}}$$

A	C	A	**G**	**C**	**T**	**T**	**C**	**T**	T	G	C

A	2	0	0	0	0	3
T	2	0	**6**	**6**	0	**3**
G	**0**	0	0	0	4	0
C	2	**6**	0	0	**2**	0
C_i	32	100	100	100	60	57

Testsequenz: 0*32+6*100+6*100+6*100+2*60+3*57 =2091

Maximalwert: 2*32+6*100+6*100+6*100+4*60+3 *57=2275

$$mat\_sim = \frac{2091}{2275} = 0.92$$

Abb. B.1.2-4 Berechnung des Matrixähnlichkeitswertes

B.2 Algorithmische und programmtechnische Umsetzung der Aufgabenstellung

Im Folgenden soll die Funktionsweise und der algorithmische Hintergrund wesentlicher Elemente der Software detailliert dargestellt werden.

B.2.1 Wahl der Werkzeuge

Betrachtet man das Anforderungsprofil an die erstellte Software, so bietet sich in Bezug auf den Punkt schnelle Anwendungsentwicklung für Windows-Betriebssysteme verbunden mit leistungsfähiger Datenbankschnittstelle das visuelle Entwicklungstool Microsoft Visual Basic (VB) an. In der verwendeten Version 6.0 unterstützt es nahezu alle modernen Programmiertechniken, wie die objektorientierte Softwareentwicklung. Leider ist die Geschwindigkeit von Zeichenkettenoperationen für viele der zu implementierenden Algorithmen, wie. zB. Sequenzalignment, nicht ausreichend. Daher wurde die Benutzerschnittstelle und das Programmframework in VB erstellt, die rechenintensiven Algorithmen jedoch in C unter Verwendung von Microsoft Visual C++ V 6.0 programmiert. Die dabei erstellte Funktionenbibliothek wurde zu einer „Dynamic Link Library" (dll-Datei) kompiliert, welche aus VB genutzt werden kann.
Als Schnittstelle zur Oracle-Datenbank wurde die MS-Active-Data-Objects (ADO) – Technologie in Verbindung mit dem entsprechenden OracleClient verwendet. Die notwendigen Daten-Auswahl und Manipulationsabfragen wurden unter Verwendung der Structured Query Language (SQL) erstellt.
Da VB nur rudimentäre Werkzeuge zur Druckausgabe bereitstellt, wurde zusätzlich der Reportgenerator Crystal Reports V 8.0 (CR) genutzt. CR ermöglicht die Definition von komplexen Berichtsvorlagen, welche z.B. auch Diagramme und Grafiken enthalten können.
Die unter Verwendung der in der Oracle-Datenbank vorliegenden Daten generierten Berichte können in einer Art Druckvorschau angezeigt werden und unter Beachtung des WYSIWYG-Prinzips ausgedruckt oder in Standardformaten (PDF,WORD, HTML,...) exportiert werden.
Zu Beginn der Arbeit wurde versucht, auf die Eigenimplementierung von Standardalgorithmen zu verzichten und statt dessen auf freie im Internet erhältliche Funktionenbibliotheken zurückzugreifen. Diese waren jedoch zumeist entweder im Funktionsumfang limitiert oder stellen ein in sich geschlossenes System dar (wie z.B. die EMBOSS-Library [Bleasby 2000]), bei dem das Herauslösen einzelner Funktionen nicht möglich ist. Zudem sind nahezu alle Bibliotheken/Programmsammlungen zur Verwendung als Kommandozeilen-Utilities auf UNIX-Systemen entwickelt. Dies gilt zwar auch für das in Hinblick auf Funktionsumfang und Modifizierbarkeit geeignete

„U.S.C. Sequence Alignment Package" [Hardy 1997], jedoch konnte dieses nach umfangreichen Anpassungsarbeiten als Dynamic-Link-Library (DLL) kompiliert und aus dem VB-Framework heraus genutzt werden. Im Laufe der Arbeiten stellten sich jedoch auch hier eine Reihe von Limitationen heraus, so dass sukzessive fast alle Alignment-Routinen selbst implementiert wurden.
Zur Unterstützung einer DNA-Motivsuche wurde die Regular-Expression-Engine von Microsoft verwendet [Hui 1999]. In IUPAC-Schreibweise (z.B. „W" entspricht „A" oder „T") eingegebene Motivausdrücke werden von einer vorgeschalteten Routine in die von der Engine benötigte Syntax umgewandelt.

B.2.2 Sequenzeditor

Zunächst wurden die von Visual Basic standardmäßig zur Verfügung gestellten Steuerelemente auf ihre Eignung als einfacher DNA-Editor hin evaluiert. Schnell wurde jedoch deutlich, dass das Textfeld-Steuerelement ebenso wie das leistungsfähigere Rich-Text-Format-Element den Anforderungen nicht gerecht werden konnten. Um Funktionen wie Darstellung der Aminosäuresequenz codierender Bereiche, geschützte Sequenzabschnitte, variable Markierungen und farbige Darstellung der Basen bzw. Codons etc. zu ermöglichen, wurde basierend auf einem Standardformular ein eigenständiger Editor entwickelt. Dazu werden die vom Formular zur Verfügug gestellten Tastatur- und Mausereignisse ausgewertet. Die DNA- und Aminosäuresequenz werden als Zeichenketten verwaltet, ebenso werden positionsabhängige Merkmale, wie vor Veränderung geschützte Bereiche, in parallel verwalteten gleichlangen Zeichenketten gespeichert. Um die programmiertechnische Komplexität zu verringern und die Geschwindigkeit der Darstellung mittels der Standard-Ausgabebefehle zufriedenstellend zu halten, wurde auf eine mehrzeilige Darstellung der Sequenz verzichtet.

B.2.3 Optimierung der kodierenden DNA-Sequenz

B.2.3.1 Grundsätzliche Funktionsweise des Optimierungsalgorithmus

Während die Optimierung eines Exons auf einen Parameter, wie z.B. optimale Kodonwahl, notfalls noch „per Hand" erfolgen kann, stellt die Ermittlung einer Sequenz, welche die Anforderungen mehrerer Kriterien möglichst optimal erfüllt, eine komplexe Aufgabe dar.

Offensichtlich wäre die ideale Lösung, sämtliche möglichen Kombinationen der für eine Aminosäuresequenz kodierenden Kodons zu bilden und die jeweils resultierende DNA-Sequenz gemäß der vorgegebenen Parameter zu bewerten. Aufgrund der enormen Anzahl möglicher Kombinationen ist dieses Verfahren jedoch bereits für relativ kurze Aminosäuresequenzen nicht mehr durchführbar (Abb. B.2.3.1-3).

Einen Ausweg bieten stochastische Verfahren, z.B. eine Variante des *„Simulated Annealing*". Der stark reduzierte Rechenaufwand durch die drastisch verringerte Anzahl evaluierter Kombinationen wird jedoch zumeist mit einer nicht-optimalen Problemlösung erkauft. Zudem muss, sofern DNA-Motive eingeführt werden sollen, aufgrund der Unstetigkeit der Problemstellung (die Änderung eines einzigen Codons kann das Vorhandensein/die Abwesenheit eines Motives bewirken), in jedem Fall vor der stochastischen Optimierung ein anderer Algorithmus innerhalb der degenerierten DNA-Sequenz potentielle Motive lokalisieren.

Viele biologische Phänomene, z.B. Erkennungsmotive für DNA-bindende Proteine, Spitzen im GC-Gehalt, Haarnadelschleifen etc. spielen sich jedoch in einem lokal stark begrenzten Bereich (<50 bp) ab, so dass ein Algorithmus wünschenswert ist, der zumindest innerhalb eines begrenzten Längenfensters die theoretisch optimale Sequenz ermitteln kann. In der vorliegenden Arbeit wurde diese Herausforderung durch die Anwendung eines „gleitenden Kombinationsfensters" gelöst.

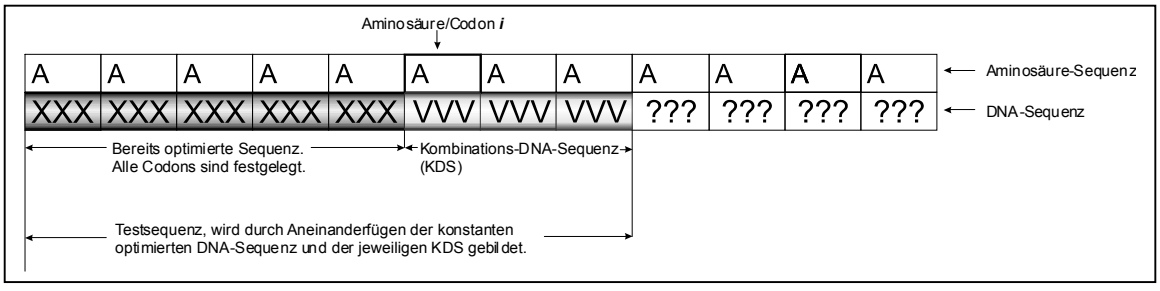
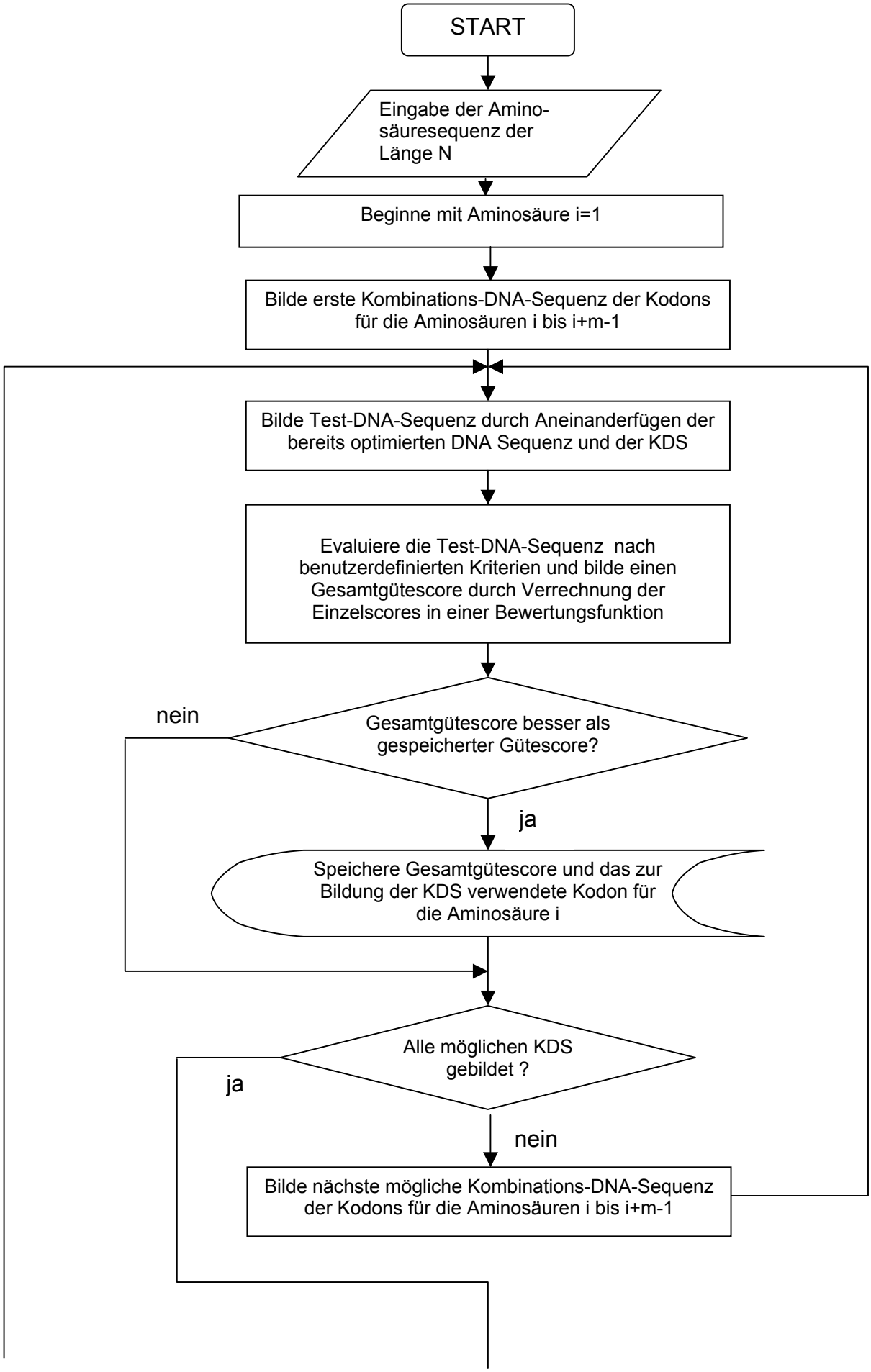


Abb. B.2.3.1-1 Definition der verwendeten Begriffe

Zur Erläuterung des Algorithmus wird die Wahl des Kodons für die *i*te Aminosäure einer Aminosäuresequenz der Länge *N* betrachtet. Dazu werden sämtliche möglichen Kodonkombinationen der verfügbaren Kodons für die Aminosäuren *i* bis *i*+*m*-1 gebildet. Jede Kombination resultiert in einer individuellen Kombinations-DNA-


START
Eingabe der Amino-
säuresequenz der
Länge N
Beginne mit Aminosäure i=1
Bilde erste Kombinations-DNA-Sequenz der Kodons
für die Aminosäuren i bis i+m-1
Bilde Test-DNA-Sequenz durch Aneinanderfügen der
bereits optimierten DNA Sequenz und der KDS
Evaluiere die Test-DNA-Sequenz nach
benutzerdefinierten Kriterien und bilde einen
Gesamtgütescore durch Verrechnung der
Einzelscores in einer Bewertungsfunktion
nein
Gesamtgütescore besser als
gespeicherter Gütescore?
ja
Speichere Gesamtgütescore und das zur
Bildung der KDS verwendete Kodon für
die Aminosäure i
Alle möglichen KDS
gebildet ?
ja
nein
Bilde nächste mögliche Kombinations-DNA-Sequenz
der Kodons für die Aminosäuren i bis i+m-1

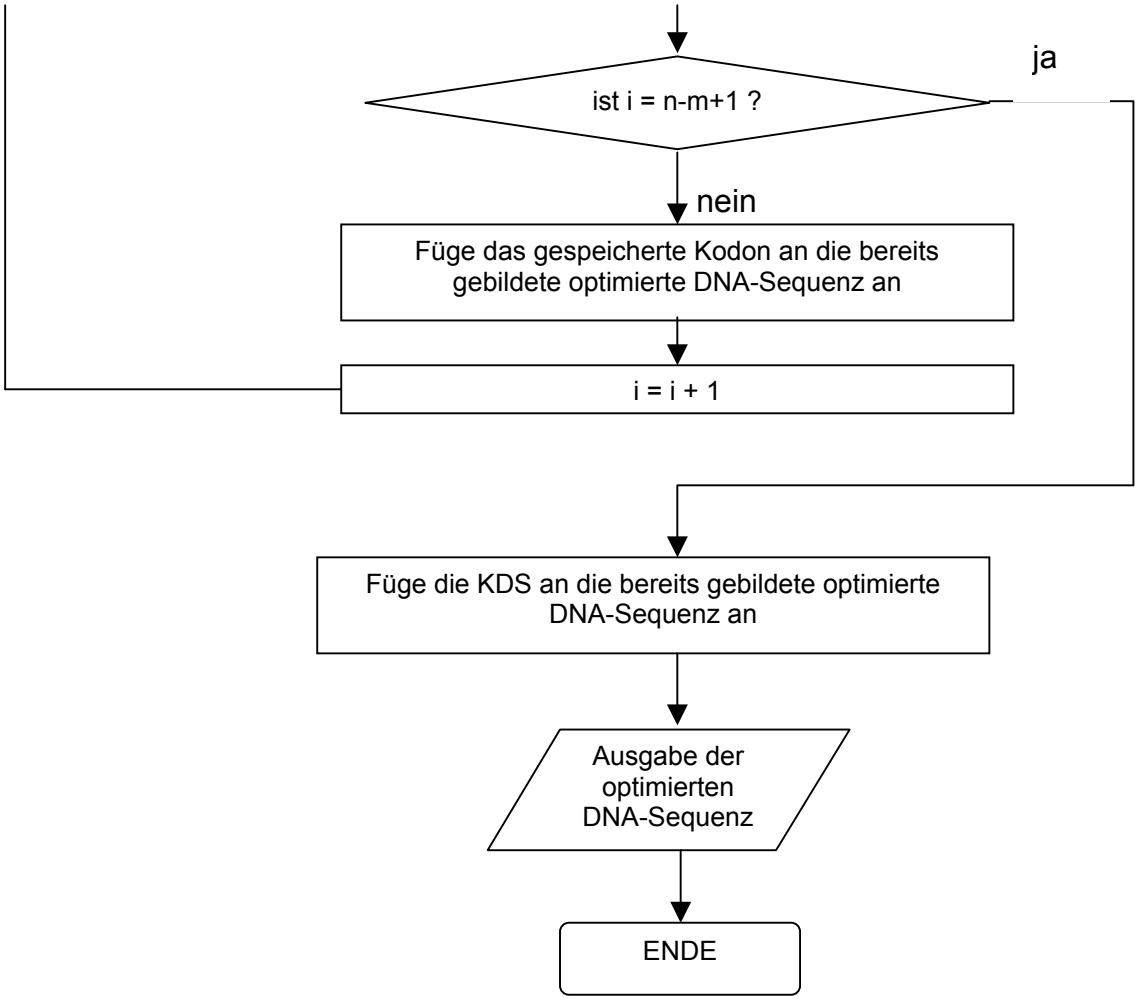


Abb. B.2.3.1-2 Flussdiagramm des Algorithmus

Sequenz der Länge 3*m* Basen, im Folgenden KDS genannt. Aus der in vorhergehenden Schritten bereits für die Aminosäuren 1 bis *i*-1 gefundenen optimalen DNA-Sequenz und jeder KDS wird durch Aneinanderhängen jeweils eine Testsequenz generiert, welche (ganz oder teilweise) nach verschiedenen Kriterien bewertet wird. Die so gefundene Güte der DNA-Sequenz für jedes individuelle Testkriterium lässt sich als numerischer Score ausdrücken. Durch Addition der nach benutzerdefinierten Vorgaben individuell mit den Faktoren G_q gewichteten Scores wird ein Gesamtscore gebildet, durch welchen die Gesamtgüte der Testsequenz beschrieben wird.

F. B.2.3.1-1

$$GesamtScore = \sum_{q=1}^{Anzahl\ der\ Kriterien} G_q * Score_q$$

Neben der individuellen Gewichtung können die Scores auch durch geeignete mathematische Funktionen modifiziert werden, wie z.B. abschnittsweise definierte

Funktionen, die die Definition von Schwellwerten gestatten (z.B. Beachtung von invers-komplementären Repetitionen erst ab einem bestimmten Alignment-Score) oder eine Nichtlinearität einführen (z.B. bei der Bewertung der Kodonwahl oder des GC-Gehalts).

An die in vorhergehenden Schritten bereits für die Aminosäuren 1 bis *i*-1 gefundene optimale DNA-Sequenz werden nun die die Aminosäuren *i* bis *i*+*k* darstellenden Kodons angefügt, welche für die Bildung der Testsequenz mit der höchsten Güte verwendet wurden. Entsprechend wird für die Ermittlung der optimalen Kodons für die Aminosäuren *i*+1 bis *N*-*m* verfahren, wobei das Kombinationsfenster um *k* Aminosäuren verschoben wird. Für die *m* Aminosäuren am Ende der Aminosäuresequenz kann direkt die komplette Kombinations-DNA-Sequenz verwendet werden, die zur besten Testsequenz geführt hat.

Der Parameter *m* kann in weiten Bereichen variiert werden. In Abb. B.2.3.1-3 ist exemplarisch die Problematik kleiner *m*'s für den Extremfall *m*=1 dargestellt (Abb. a). Hier bilden die (bezüglich des Kodongebrauchs) optimalen Kodons für die Aminosäuren *i* und *i*+1 eine Restriktionsschnittstelle, welche aber vermieden werden soll. Dies wird idealerweise durch die Wahl des nur geringfügig schlechteren zweitbesten Kodons für die Aminosäure *i* erreicht, so dass für die Aminosäure *i*+1 das optimale Kodon statt des hier wesentlich schlechteren zweitbesten Kodons genutzt werden kann. Für *m*=1 kann der Algorithmus das drohende Entstehen dieses Motivs jedoch erst im nächsten Schritt, also bei Betrachtung der Aminosäure *i*+1, erkennen, was in der Verwendung des besten Kodons für die Aminosäure *i* und des sehr schlechten Kodons für Aminosäure *i*+1 resultiert.

Im Sinne einer bestmöglichen Optimierung muss also eine möglichst hohe Zahl variierter Kodons angestrebt werden, d.h. der Algorithmus soll vom aktuell bearbeiteten Kodon *i* möglichst weit „in die Zukunft" blicken können. Da mit wachsendem *m* die Zahl der zu evaluierenden Testsequenzen exponentiell ansteigt, sind bei der für aktuelle Personalcomputer verfügbaren Rechenleistung Werte zwischen 5 und 10 sinnvoll.

Auch die Zahl der pro Schritt festgelegten Kodons *k* kann variiert werden. Im Grenzfall *l*=*m* werden also alle variierten Kodons nach Emittlung der optimalen Testsequenz festgelegt und das Fenster zur Aminosäure *i*+*k* versetzt. Jedoch tritt hierbei an den Grenzen der „Kombinationsblöcke" dieselbe Problematik wie bereits für den Fall *m*=1 auf, so dass im Idealfall *k*=1 gesetzt wird (Abb. B.2.3.1-3d).

Eine deutliche Beschleunigung des Algorithmus kann erreicht werden, wenn zunächst die Testsequenz nach den Kriterien evaluiert wird, die einen positiven Beitrag zur Gütefunktion liefern (z.B. die Kodonwahl), und anschließend die Kriterien, die einen negativen Beitrag liefern (z.B. Repetitionen). Idealerweise werden bei letzteren die

Bewertungen zuerst vorgenommen, welche nur einen geringen Aufwand an Rechenzeit erfordern. Dadurch wird ermöglicht, dass zu Beginn bzw. nach jeder Evaluierung eines „Negativ"-Kriteriums der Wert der Gütefunktion durch Summation der (soweit festgestellten) Einzelscores berechnet und mit dem Gütescore der besten bislang getesteten Testsequenz verglichen werden kann. Ist der Gütescore der momentan evaluierten Testsequenz bereits schlechter, kann auf die weitere Evaluierung verzichtet werden und die nächste KDS/Testsequenz gebildet werden.
Prinzipiell lassen sich die meisten Kriterien als negative Scores ausdrüchen, sofern ein zu erwartender Maximalwert festgelegt werden kann. In diesem Fall kann die Differenz zum Maximalwert von der Gütefunktion abgezogen werden.

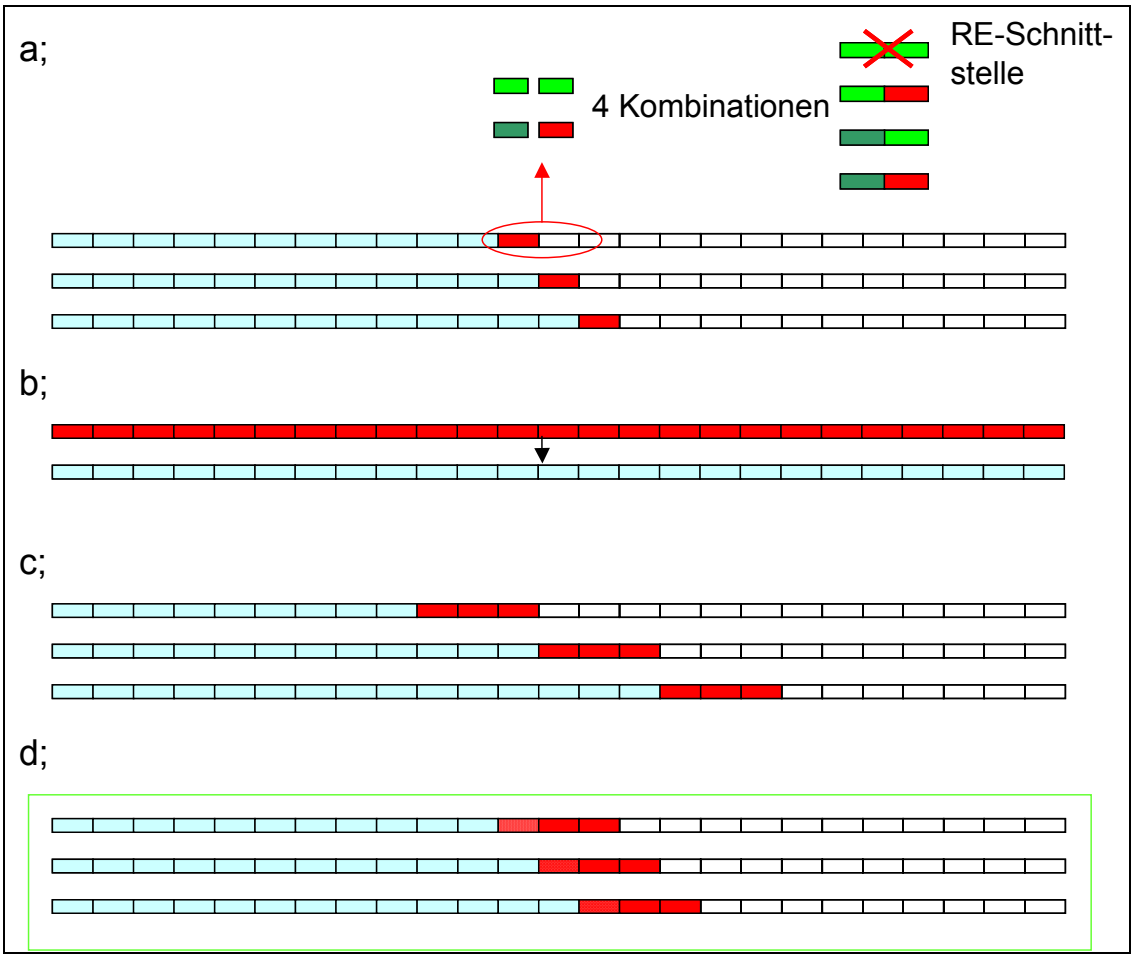


Abb. B.2.3.1-3 Verschiedene Möglichkeiten der deterministischen Sequenzoptimierung: a; Kodon für Kodon. Die Entstehung einer Restriktionsschnittstelle kann nur durch Nutzung eines sehr schlechten Kodons vermieden werden. B; Durchtesten aller möglichen Kombinationen und Festlegen der Besten. C; Kombinationsfenster von 3 Kodons Länge, das nach Festlegen der besten Dreierkombination um 3 Kodons weiter geschoben wird. D; Dreier-Kombinationsfenster. Jedoch wird nur ein Kodon der besten Kombination festgelegt und das Fenster wird um ein Kodon weitergeschoben.

Zur Evaluierung der Testsequenzen können grundsätzlich beliebige Kriterien verwendet werden. Die wichtigsten und von der vorliegenden Software berücksichtigten werden nachfolgend erläutert.

B.2.3.2 Anpassen der Kodonwahl

Die Anpassung des Kodongebrauchs des synthetischen Genes zur Steigerung der Expression ist eines der wichtigsten Kriterien bei der Optimierung. Um die Güte eines Kodons im gewählten Expressionssystem abzuschätzen, analysiert man die Kodonwahl von (idealerweise hochexprimierten) Genen des Wirtsorganismus, welcher in der Regel mit den entsprechenden tRNA-Frequenzen korreliert. Als Rohdaten erhält man dabei C_{ij} als Häufigkeit der Verwendung des Kodons *j* für die Aminosäure *i* in den untersuchten Genen. Bezogen auf die Summe der n_i für die Aminosäure *i* kodierenden Kodons ergibt sich die *Codon Usage* CU_{ij}.
Diese Kodonfraktionen sind für eine Vielzahl von Organismen tabelliert und können über das Internet (z.B. die Kazusa Codon Usage Database auf http://www.kazusa.or.jp/codon/) abgerufen werden [Nakamura 2000].

F. B.2.3.2-1

$$CU_{ij} = \frac{C_{ij}}{\sum_{j=1}^{n_i} C_{ij}}$$

Bei einer einparametrigen Optimierung auf Kodongebrauch kann C_{ij} direkt zur Wahl des für eine Aminosäure optimalen Kodons verwendet werden. Sobald jedoch ein weiteres Kriterium berücksichtigt werden soll, müssen auch Kodons unterschiedlicher Aminosäuren in ihrer Güte miteinander verglichen werden können.

Aminosäure	Kodon	*Codon Usage* [%]	$\boldsymbol{w_{ij}}$*100	*RSCU*
Glutamat	GAA	60	100	1,2
	GAG	40	67	0,8
Leucin	CTG	25	100	1,5
	TTG	20	80	1,2
	TTA	20	80	1,2
	CTT	18	72	1,1
	CTC	16	64	1
	CTA	1	4	0,1

Abb. B.2.3.2-1 Vergleich der Bewertungsmöglichkeiten für die Kodongüte

Wie in Abb. B.2.3.2-1 an einem einfachen Zahlenbeispiel ersichtlich ist, können die Fraktionsangaben hierfür nicht direkt verwendet werden, da hierbei z.B. bei direkten Vergleich der Prozent-Zahlenwerte das für Leucin optimale Kodon *CTG* mit 25% Anteil wesentlich schlechter bewertet würde, als das zweitbeste Glutamat-Kodon *GAG* mit 40% Anteil, obwohl die tatsächlichen Verhältnisse natürlich genau umgekehrt sind.

Um die von Aminosäure zu Aminosäure unterschiedliche Anzahl n_i synonymer Kodons zu berücksichtigen, kann man sich der „*Relative Synonymous Codon Usage*" ($RSCU_{ij}$) bedienen, unter der man das Verhältnis zwischen beobachteter *Codon Usage* und bei gleichhäufiger Nutzung der synonymen Kodons erwarteten *Codon Usage* versteht [Mathur 1991].

F. B.2.3.2-2

$$RSCU_{ij} = \frac{CU_{ij}}{1/n_i}$$

Alternativ kann näherungsweise die „relative Angepasstheit" („relative adaptiveness") w_{ij} eines Kodons verwendet werden, worunter man das Verhältnis des Anteils eines bestimmten Kodons zur Fraktion des am häufigsten verwendeten synonymen Kodons versteht:

F. B.2.3.2-3

$$w_{ij} = \frac{C_{ij}}{C_{ij\max}}$$

Wenngleich in Bezug auf die Zahl der synonymen Kodons die RSCU-Werte statistisch exakter sind, spiegelt die relative Angepasstheit die biologische Realität wohl besser wieder.

Bildet man das geometrische Mittel der *w*-Werte der *L* betrachteten Kodons, so erhält man den *Codon Adaption Index CAI*, welcher in der Literatur zur Vorhersage der Expressionsstärke eines gegebenen Genes verwendet wird [Sharp 1987].

F. B.2.3.2-4

$$CAI = \left(\prod_{k=1}^{L} w_k \right)^{1/L}$$

Dieser kann beispielsweise als *CUScore* zur Bewertung einer Testsequenz herangezogen werden, wobei die Berechnung des *CAI* für die KDS ausreichend ist. In allen gezeigten Beispielen ist der *Codon-Usage*-Score *CUScore* jedoch als das arithmetische Mittel der *w*-Werte multipliziert mit 100 definiert:

F. B.2.3.2-5

$$CUScore = \frac{\sum_{k=1}^{L} w_k}{L} * 100$$

B.2.3.3 Berechnen von gemischten Codon Usage Tabellen

Wenngleich die Kodonwahl eines synthetischen Genes idealerweise auf die eines ganz bestimmten Expressionssystems angepasst wird, kann es manchmal wünschenswert sein, das synthetisiserte Gen so zu entwerfen, dass es auch in mehreren verschiedenen Organismen mit akzeptablen Ausbeuten exprimiert werden kann. Dies gilt insbesondere, wenn die Herstellung mehrerer hinsichtlich ihrer Kodonwahl auf jeden der Organismen jeweils optimierten Gene aus ökonomischen oder zeitlichen Gründen nicht möglich ist.
In diesem Fall müssen bei der Bewertung eines Kodons zwei Aspekte berücksichtigt werden:

Ein Kodon ist umso besser,

- je größer *w* in beiden Organismen A und B ist.
- je kleiner die Differenz zwischen w_A und w_B ist.

Die Güte eines Kodons bei Verwendung in zwei verschiedenen Expressionssystemen kann also folgendermaßen ausgedrückt werden:

F. B.2.3.3-1

$$Z = w_A + w_B - \left| w_A - w_B \right|$$

Beispielsweise ist also die Verwendung eines Kodons, welches in beiden Organismen die relative Adaptiveness w =0.5 aufweist, besser, als ein Kodon mit den Werten 0.1 und 1.0 . Dies entspricht der Absicht, bei der gemeinsamen Optimierung auf die beiden Organismen besonders solche Kodons zu vermeiden, die in einem der Organismen kaum verwendet werden und daher die Expression des synthetischen Gens stark limitieren würden. Aus den Z-Werten kann dann eine „Kompromiss-*Codon-Usage*-Tabelle" errechnet werden, welche für die Optimierung verwendet wird. Soll eine solche für mehr als zwei Organismen erstellt werden, werden zur Berechnung des Z-Wertes für ein gegebenes Kodon zunächst alle *w*-Werte aufsummiert. Die *w*-Werte werden sortiert und von der Gesamtsumme jeweils die Differenz eines *w*-Wertes zum nächstgrößeren abgezogen.

B.2.3.4 Anpassen des GC-Gehalts

Der GC-Gehalt ist eine wichtige Kenngröße einer DNA-Sequenz und spielt sowohl im Expressionssystem wie auch bei der Herstellung synthetischer Gene eine wichtige Rolle.

Ein zu hoher GC-Gehalt (>80 %) kann zur genetischen Instabilität, z.B. in *E. coli,* führen und erschwert die Synthese des Gens, indem z.B. das Risiko von Fehlhybridisierungen erleichtert wird. Andererseits führt auch ein zu niedriger GC-Gehalt zu Problemen, z.B. beim Aufbau der Gene durch Ligation oder PCR, da für eine akzeptable Hybridisierungstemperatur sehr lange Hybridisierungsstrecken notwendig sind. Ein stellenweise zu niedriger GC-Gehalt kann ebenso zu unzureichender genetischer Stabilität führen, wie von Lee et al am Beispiel der Inserts eines Polio-Vaccinevektors gezeigt. Die teilweise geringe genetische Stabilität der GC-armen Wildtyp-Inserts konnte durch die Einführung mehrerer stiller Mutationen, welche den GC-Gehalt erhöhten, dramatisch gesteigert werden [Lee 2002].

Bei der optimierten Sequenz ist sowohl eine möglichst geringe Abweichung des durchschnittlichen Gesamt-GC-Gehaltes vom vorgegebenen Wunsch-GC-Gehalt erforderlich, ebenso müssen aber vor allem auch Schwankungen über den Verlauf der Sequenz (z.B. GC-Gehalts-Spitzen) gering gehalten werden. Wie gut diese Anforderungen erfüllt werden, lässt sich am besten mittels eines Graphen des gleitenden GC-Durchschnitts beurteilen. Dazu wird für jede Sequenzposition *k* der durchschnittliche GC-Gehalt der Basen von *k-b* bis *k+b*, wobei *b* im Bereich von 10 bis 20 Basen liegt, errechnet und gegen die Position aufgetragen.

Zur Evaluierung einer Testsequenz kann der Absolutwert der Differenz des durchschnittlichen GC-Gehaltes der letzten 20 bis 30 Basen der Testsequenz und des gewünschten GC-Gehaltes herangezogen werden. Je kleiner die Differenz, umso besser ist die Testsequenz.

F. B.2.3.4-1

$$GCScore = -\left|\langle GC\rangle - GC_{Wunsch}\right|$$

Einen hohen Anteil am GC-Score haben jedoch Basen, welche durch unterschiedliche KDS nicht geändert werden können, wie z.B. die in die Berechnung des durchschnittlichen GC-Gehalts miteinbezogenen Nucleotide vor der KDS und die innerhalb synonymer Kodons unveränderlichen Nucleotide. Um dies zu veranschaulichen, werden zwei DNA-Sequenzen Seq1 und Seq2 betrachtet, welche in vorhergehenden Optimierungsschritten festgelegt wurden. Beide sollen nun durch die optimale DNA-Sequenz für die Aminosäuresequenz *RFIRK* ergänzt werden.

Der GC-Gehalt wird als Durchschnitt der 20 letzten Basen der Sequenzen Seq1 und Seq2 (unterstrichen gezeigt) sowie der KDS für die Aminosäuresequenz ermittelt.

Angestrebt wird ein GC-Gehalt von $\%GC_w$=60%. Da Seq2 im Gegensatz zu Seq1 sehr GC-arm ist, sollen die Unterschiede der möglichen KDS hinsichtlich des GC-Gehaltes im Vergleich zu anderen Optimierungskriterien wie Kodonwahl bei Seq2 stärker berücksichtigt werden, als bei Seq1, deren GC-Gehalt der Zielvorgabe bereits nahe kommt. Dies kann, wie allgemein in F. B.2.3.4-2 definiert, durch eine Potenzierung, beispielsweise eine Quadrierung des über F. B.2.3.4-1 definierten GCScore, erreicht werden.

Exemplifiziert wird dies durch Betrachtung der für die Aminosäuresequenz möglichen KDS mit höchstem (KDS 2) und mit niedrigstem (KDS 1) GC-Gehalt (Abb. B.2.3.4-1). Die Tabelle Abb. B.2.3.4-2 stellt die 4 möglichen Kombinationen der zwei Sequenzen mit den beiden KDS dar. Entscheidend für die Gewichtung des GC-Score im Verhältnis zur Gewichtung anderer Kriterien ist der **Unterschied** im GC-Score, welcher durch die Verwendung der beiden KDS erreicht werden kann. Wird $\Delta(\%GC)$ als GC-Score verwendet, so ist der **Unterschied** in der Bewertung der beiden KDS gleich (jeweils 20), unabhängig ob die Testsequenz mit der GC-reichen Sequenz Seq1 oder der GC-armen GC-Sequenz Seq2 gebildet wird. Wird dagegen $\Delta(\%GC)^2$ verwendet, so hängt der Unterschied auch von den lokalen unveränderlichen Eigenschaften (hier GC-Gehalt) der Testsequenzen ab, so dass die Gewichtung des GC-Gehaltes stärker in die Gesamtgütefunktion einfließt.

F. B.2.3.4-2

$$GCScore = -\left|\langle GC\rangle - GC_{Wunsch}\right|^{p}$$

DNASeq1:

GCCGTGGCCGACCCTGGTGACCACCTTTACCTATGGCGTGCAG

11 GC's in den letzten 20 Basen

DNASeq2:

GGATAAACAGAAAAACGGCATTAAAGTGAACTTTAAAATTCGC

6 GC's in den letzten 20 Basen

	R__	F__	I__	R__	K__	
KDS1:	AGA	TTT	ATT	AGA	AAA	2 GC
KDS2:	CGG	TTC	ATC	CGG	AAG	9 GC

Abb. B.2.3.4-1 Beispiel zur nichtlinearen Gewichtung des GC-Gehaltes

	KDS1			KDS2				
	%GC	$\Delta(\%GC)$	$\frac{\Delta(\%GC)^2}{10}$	%GC	$\Delta(\%GC)$	$\frac{\Delta(\%GC)^2}{10}$	$\Delta\Delta(\%GC)$	$\frac{\Delta\Delta(\%GC)^2}{10}$
DNASeq1 (55% GC in den letzten 20 Basen)	37,1	22,9	52,4	57,1	2,9	0,8	20	52
DNASeq2 (30% GC in den letzten 20 Basen)	22,8	37,2	138,3	42,9	17,1	29,2	20	109

Abb. B.2.3.4.-2 Mögliche GC-Scores bzw. Scoreunterschiede bei unterschiedlichen Kombinationen DNA-Sequenz/KDS. Dabei bedeuten

%GC: *Der GC-Gehalt der letzten 35 Basen der gebildeten Testsequenzen*

$\Delta(\%GC)$: *$\%GC_w$ - %GC, als GC-Score verwendbar*

$\Delta(\%GC)^2$: *$(\%GC_w - \%GC)^2$, als GC-Score verwendbar*

$\Delta\Delta(\%GC)$: *Der maximal erreichbare Scoreunterschied bei Verwendung der beiden unterschiedlichen KDS mit linearer $\Delta(\%GC)$ Gewichtung*

$\Delta\Delta(\%GC)^2$: *Der maximal erreichbare Scoreunterschied bei Verwendung der beiden unterschiedlichen KDS mit quadratischer $\Delta(\%GC)$ Gewichtung*

B.2.3.5 Einführen bzw. Ausschließen von DNA-Sequenzmotiven

Lokale Erkennungssequenzen bzw. biophysikalische Charakteristika spielen in der gesamten Zell- und Molekularbiologie eine entscheidende Rolle. Es ist offensichtlich, dass eine unbeabsichtigte Generierung solcher Motive innerhalb der kodierenden Sequenz des synthetischen Genes eine für den Wirtsorganismus toxische Wirkung haben und die Expression stark reduzieren oder ganz unterdrücken kann [Bieth 1997]. Bei der Optimierung ist es daher entscheidend, die unbeabsichtigte Generierung solcher Motive auszuschließen. Umgekehrt kann es durchaus wünschenswert sein, bestimmte Basenmotive in die codierende Sequenz einzuführen, z.B. Restriktionsenzym-Schnittstellen oder immunomodulatorische CpG-Motive.

Die im Rahmen dieser Arbeit entwickelte Optimierungssoftware bietet zwei verschiedene Möglichkeiten der Motivsuche.

Zum einen wurde basierend auf dem Regular-Expression-Automatisierungsobjekt von Microsoft eine IUPAC-Consensussuche implementiert. Dazu werden die vom Anwender bereitgestellten IUPAC-Consensi, wie z.B. YYGCN(1;3)GAA, zunächst in die von der Microsoft-Komponente benötigte Syntax übersetzt. Zur Beurteilung einer Testsequenz wird diese oder zumindest die letzten *k* Basen, mit *k* größer gleich der

Länge des längsten DNA-Motives, auf das Vorkommen der auszuschließenden bzw. einzuführenden DNA-Motive hin untersucht. Aus der von der Komponente zurückgegebenen Anzahl der gefundenen Motive n_i kann nach Multiplikation mit einem Gewichtungsfaktor g_i, welcher positiv für einzuführende Motive und negativ für auszuschließende Motive ist, ein Gütescore für die Testsequenz ermittelt werden. Dieser ist folglich umso größer (besser), je mehr einzuführende und je weniger auszuschließende Motive in der Testsequenz enthalten sind.

F. B.2.3.5-1

$$SiteScore_{IUPAC} = \sum_{i=1}^{\substack{Anzahl \\ Motive}} g_i * n_i$$

Zu den bereits in Kap. B.1.2 angesprochenen Nachteilen der IUPAC-Consensusdarstellung kommt hinzu, dass die Komponente keine fehlertolerante Suche, wie z.B. „Suche alle Sites, welche eine 90%ige Übereinstimmung mit der Consensus Sequenz YYGCN(1;3)GAA aufweisen" zulässt. Daher ist die Gefahr, bei einer restriktiv formulierten Consensus-Sequenz eine real vorhandene Bindungsstelle zu übersehen, sehr groß.

Aus diesem Grund unterstützt die Software auch die Erkennung von als Nucleotid-verteilungsmatrix definierten DNA-Motiven. Im Gegensatz zur IUPAC-Consensus-Suche liefert eine auf Verteilungsmatrizes basierende Untersuchung der Testsequenz keine Ja/Nein Antwort, sondern erkannte Sites werden durch einen Score *F* charakterisiert, welcher die Ähnlichkeit der gefundenen potentiellen Bindungsstelle mit der durch die Verteilungsmatrix definierten optimalen Bindungsstelle darstellt. Dieser Score steht in direkter Beziehung zur Erkennungssicherheit (d.h. eine potentielle Bindungsstelle ist im biologischen System funktionell) bzw. biologischen Aktivität (d.h. je höher der Score, desto stärker die biologische Aktivität). Daher können zum einen durch die Definition eines Cutoff-Scores nur solche Sites in die weitere Betrachtung mit einbezogen werden, die mit hinreichender Sicherheit keine falsch-positiven Fundstellen darstellen. Zum anderen kann durch die Bildung des Testsequenz-Gütescores aus der Summe der Fundstellen-Einzelscores $F_{i,j}$ die Testsequenz viel diffiziler bewertet werden.

F. B.2.3.5-2

$$SiteScore_{Matrix} = \sum_{i=1}^{Anzahl\ Motive} g_i * \sum_{j=1}^{Anzahl\ Fundstellen} F_{i,j}$$

Da durch die Verwendung des Consensus-Vektors den verschiedenen Basen einer Erkennungssequenz eine unterschiedlich starke Bedeutung für die Erkennung/biologische Aktivität zugeordnet wird, kann bei der durch das geschilderte Optimierungsverfahren idealerweise betriebenen vollständigen Durchtestung des KDS-Kombinationsraumes die Sequenz gefunden werden, die z.B. ein DNA-Motiv durch Eliminierung der für die Aktivität wichtigsten Basen am effektivsten ausschaltet (bzw. eine ideale Kompromisslösung bei Einbeziehung anderer Kriterien gefunden werden).

B.2.3.6 Ausschließen repetitiver Elemente

Auch stark repetitive Sequenzabschnitte zeichnen sich u.U. durch eine geringe genetische Stabilität aus. Darüber hinaus ist einleuchtend, dass repetitive Sequenzen zu teilweise homologen Oligonucleotiden führen, wodurch vor allem bei der Batch-Synthese eine stark erhöhte Gefahr von Fehlhybridisierungen besteht. In die Bewertung der Testsequenz muss deshalb auch einfließen, ob diese identische oder einander ähnliche Sequenzabschnitte enthält. Dies kann durch die Berechnung des optimalen Alignmentscores eines lokalen Alignments der Testsequenz mit sich selbst leicht festgestellt werden.

Da nur die KDS verändert werden kann, ist es ausreichend, die KDS sowie einen 10-20 Nucleotide langen Bereich davor (um bei ausgedehnten Abschnitten mit allerdings relativ gering ausgeprägter Ähnlichkeit zueinander noch einen entsprechend hohen Alignmentscore erreichen zu können) mit der Testsequenz zu alignen.

Dies ist in Abb B.2.3.7-1 illustriert. Daraus wird deutlich, dass sowohl zueinander ähnliche Sequenzen, welche beide in dem o.g. Bereich liegen, erkannt werden können, als auch z.B. die Änlichkeit der KDS mit einem am Anfang der Testsequenz liegenden Bereich. Idealerweise errechnet sich ein Repetitivitätsscore für die Testsequenz auf Grundlage einer Summation der über einem Schwellwert liegenden Alignmentscores aller optimalen und suboptimalen Aligments. Um die Ausführungsgeschwindigkeit der Optimierung hoch zu halten, wird bei der Berechnung des Repetitivitätsscore in der vorliegenden Softwareversion jedoch nur der Alignmentscore des optimalen Alignments berücksichtigt.

Analog zur Betrachtung des GC-Gehaltes ist auch hier der durch Variation der KDS maximal erreichbare Unterschied in der Repetitivität entscheidend. Aus diesem Grund empfiehlt es sich auch hier, den Alignmentscore zu potenzieren.

F. B.2.3.6-1

$$RepScore = -Alignmentscore_{\max}^{\ p}$$

B.2.3.7 Ausschließen von Sekundärstrukturen

Die potentielle Bildung von sehr stabilen Sekundärstrukturelementen auf RNA-Ebene oder cruciformer Strukturen auf DNA-Ebene lässt sich an der Testsequenz durch das Vorhandensein invers-komplementärer Repetitionen erkennen. Cruciforme Strukturen auf DNA Ebene können die Transkription behindern und führen möglicherweise zu genetischer Instabilität [Bagga 1990, Brahmachari 1991], auf RNA-Ebene kann vermutet werden, dass sich eine stabile Sekundärstruktur negativ auf die Translationseffizienz auswirkt [Griswold 2003]. Besonders ungünstig dürften sich hierbei nahe beieinander liegende invers komplementäre Repetitionen auswirken, die stabile Haarnadelschleifen (bzw. cruciforme Strukturen) ausbilden können. Auch in der Produktion der synthetischen Gene können invers-komplementäre Repetitionen durch Fehlhybridisierungen oder die Bildung von Haarnadelschleifen im Oligo problematisch sein.

Die Überprüfung auf inverse Repetitionen kann prinzipiell analog zur Überprüfung auf Repetitionen durchgeführt werden, jedoch wird hierbei die Testsequenz mit der invers komplementären Endbereich der Sequenz aligned. Um die Stabilität einer potentiellen Sekundärstruktur näherungsweise in den Alignmentscore einfließen zu lassen, wird hierbei mit einer Scoring-Matrix gearbeitet, welche C:C und G:G – Entsprechungen im Alignment doppelt so hoch gewichtet wie A:A und T:T Paarungen.

F. B.2.3.7-1

$$SekScore = -Alignmentscore_{\max}^{\ p}$$

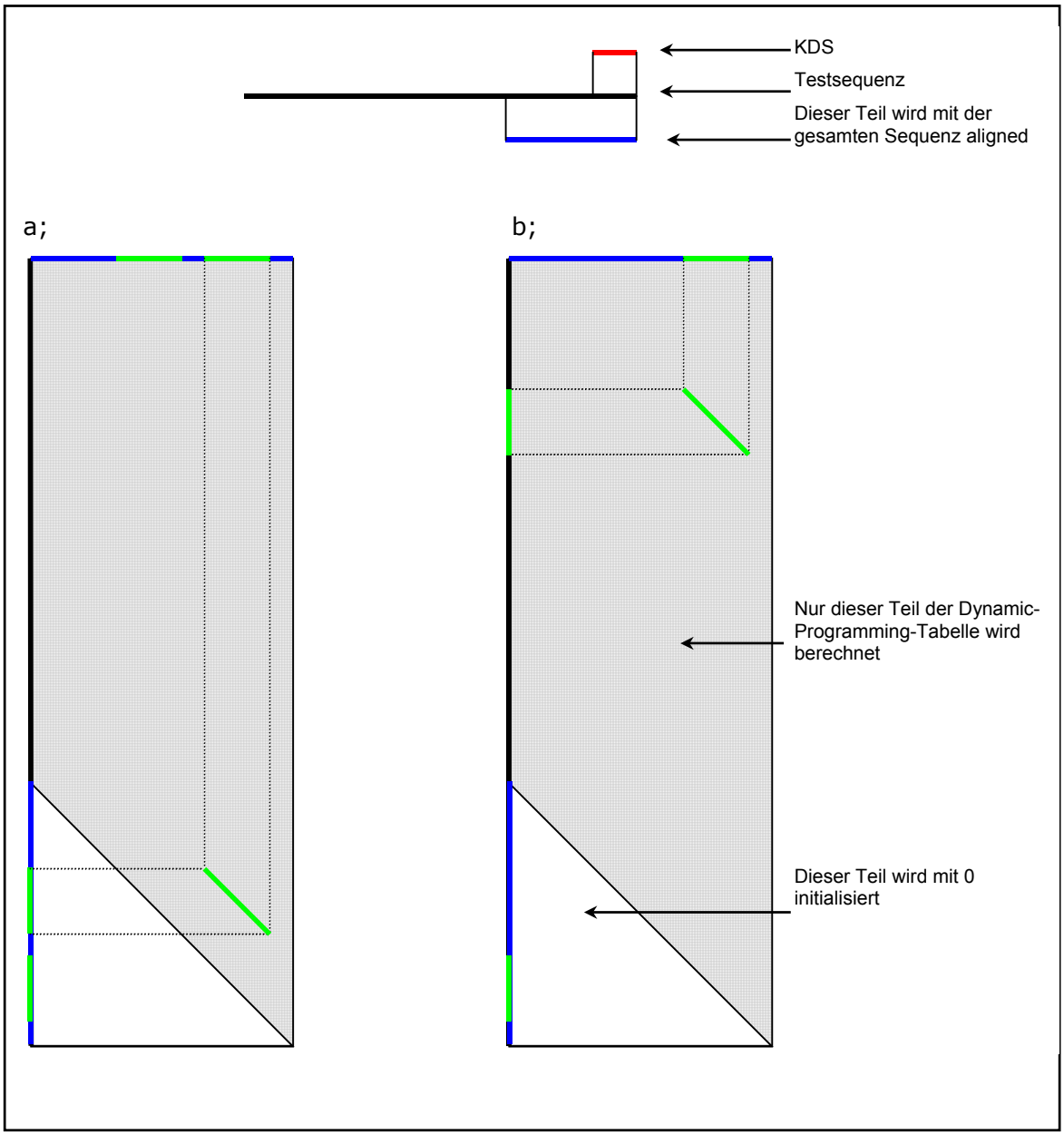


Abb. B.2.3.7-1 Durch Bildung eines lokalen Alignments der Testsequenz mit dem Ende der Testsequenz können ähnliche Bereiche (grün gekennzeichnet) zuverlässig erkannt werden. a; Die einander ähnlichen Sequenzbereiche befinden sich beide am Ende der Testsequenz b; Die einander ähnlichen Sequenzbereiche befinden sich im vorderen Teil bzw. im endständigen Teil der Testsequenz

B.2.3.8 Eleminieren starker Homologien zu einer vorgegebenen Sequenz

Beispielsweise im Bereich der DNA-Vakzinierung ist es entscheidend, dass die für die Vakzinierung verwendeten Sequenzen keine signifikante Ähnlichkeit mit den pathogenen Elementen des natürlichen Virusgenoms aufweisen, um unerwünschte Rekombinatiosereignisse sicher auszuschließen. Das eventuelle Vorkommen entsprechender Homologien wird ähnlich den obengenannten Verfahren dadurch

überprüft, dass der Alignmentscore des optimalen lokalen Alignments eines endständigen Bereichs der Testsequenz mit einer vorgegebenen Vergleichssequenz errechnet wird. Dieser fließt dann als negativer Beitrag in die Berechnung des Gesamtscores einer Testsequenz mit ein.

F. B.2.3.8-1

$$HomoScore = -Alignmentscore_{\max}{}^{p}$$

B.2.3.9 Beispielhafte Anwendung des Algorithmus

Zur Verdeutlichung der Arbeitsweise des Algorithmus soll dieser auf eine kurze Aminosäuresequenz (ASSeq) angewendet werden und die zugehörige optimale DNA-Sequenz ermittelt werden. Als Referenz dient eine konventionelle Rückübersetzung mit Optimierung auf optimale Kodonwahl.

```
 1   2   3   4   5   6   7   8   9  10  11   12  13  14
 E__ Q__ F__ I__ I__ K__ N__ M__ F__ I__ I__ K__ N__ A__

 GAA CAG TTT ATT ATT AAA AAC ATG TTT ATT ATT AAA AAC GCG
 GAG CAA TTC ATC ATC AAG AAT     TTC ATC ATC AAG AAT GCC
             ATA ATA                 ATA ATA         GCA
                                                     GCT
```

Abb. B.2.3.9-1 Die zu optimierende Aminosäuresequenz. Gezeigt sind alle für die jeweilige Aminosäure kodierenden synonymen Kodons

Folgende Parameter werden der Optimierung zugrunde gelegt
- Optimierung auf die Kodonwahl von E. Coli K12
- Wunsch-GC-Gehalt 50%
- Repetitionen sollen möglichst ausgeschlossen werden
- Die Nla III Erkennungssequenz CATG soll ausgeschlossen werden

Als Bewertungsfunktion wird folgende Funktion verwendet:

F. B.2.3.9-1 $GesamtScore = CUScore + GCScore + REPScore + SiteScore$

Der CUScore errechnet sich als das arithmetische Mittel der Relative-Adaptiveness-Werte multipliziert mit 100.

F. B.2.3.9-2 $CUScore = \langle w \rangle * 100$

F. B.2.3.9-3 $GCScore = \left| \langle GC \rangle - GC_{Wunsch} \right|^{1.3} \times 0.8$

Zur Ermittlung des Alignmentscore wurde ein lokales Alignment der Testsequenz, maximal jedoch die letzten 36 Basen mit der kompletten Testsequenz durchgeführt. Alignment-Parameter: Match = 1; Mismatch = -2; Gap = -2

F. B.2.3.9-4 $REPScore = -Alignmentscore_{\max}^{1,3}$

Für jede gefundene CATG-Sequenz wird ein Sitescore von 100000 vergeben.
Die KDS-Länge beträgt 9 Basen, d.h. es werden 3 Kodons variiert.

Eine Optimierung auf optimale Kodonwahl resultiert in folgender Sequenz:

```
1   2   3   4   5   6   7   8   9  10  11   12  13  14
E__ Q__ F__ I__ I__ K__ N__ M__ F__ I__ I__ K__ N__ A__
GAA CAG TTT ATT ATT AAA AAC ATG TTT ATT ATT AAA AAC GCG
```

Sie ist durch folgende Eigenschaften charakterisiert:
- Stark repetitiv, verursacht durch die zweimalig erscheinende Aminosäuresequenz F__I__I__K__N (gezeigt ist das repetitive Element mit dem höchsten Score (18)):

```
19 AACATGTTTATTATTAAAAAC
   |||| ||||||||||||||||
 2 AACA-GTTTATTATTAAAAAC
```

- GC-Gehalt: 21,4 %
- Die Nla III Erkennungssequenz CATG ist vorhanden
- Durchschnittliche *Codon Usage*: 100

Wird die Optimierung nach dem beschriebenen Algorithmus vorgenommen, so erhält man folgende DNA-Sequenz:

```
1   2   3   4   5   6   7   8   9  10  11   12  13  14
E__ Q__ F__ I__ I__ K__ N__ M__ F__ I__ I__ K__ N__ A__
GAA CAG TTC ATC ATC AAA AAT ATG TTT ATT ATC AAG AAC GCG
```

Sie ist durch folgende Eigenschaften charakterisiert:
- Kaum repetitiv (gezeigt ist das repetitive Element mit dem höchsten Score (6)):

```
11 TCATCA
   ||||||
 8 TCATCA
```

- GC-Gehalt: 31,0 %
- Die Nla III Erkennungssequenz CATG ist vermieden worden
- Durchschnittliche *Codon Usage*: 88

Wie zu erkennen ist, wurde an fünf Aminosäure-Positionen nicht das hinsichtlich *Codon Usage* optimale Kodon gewählt. Statt dessen stellt die gefundene Sequenz die optimale Balance der unterschiedlichen Anforderungen in Bezug auf *Codon Usage*, GC-Gehalt und ideale Sequenzeigenschaften (Vermeidung von Repetitionen) dar.
Bei den Aminosäuren Nr. 3,4,5 ist der höhere GC-Anteil der „schlechteren" Kodons der Grund für die Wahl. An Position 6 überwiegt jedoch beim Vergleich der Kodons AAA und AAG die wesentlich bessere CU des AAA Kodons, obwohl die Wahl des AAG Kodons zu einem besseren GC-Score führen würde.
Bei Bildung der KDS an Basenposition 13 wird für die Aminosäure Nr. 7 noch das Kodon AAC bevorzugt, da bei einer Fenstergröße für die KDS von 3 Kodons noch nicht erkennbar ist, dass diese Wahl zu Bildung des zu vermeidenden DNA-Motivs CATG führen wird (Für Methionin steht nur ein Kodon zur Verfügung!). Bei der Bildung der KDS an Basenposition 16 wird dies jedoch bereits erkannt und folgerichtig das Kodon AAT gewählt.
Bei der Wahl der Kodons für die Aminosäuren 9-13 spielt neben Codon Usage und GC-Gehalt auch die Vermeidung einer repetitiven DNA-Sequenz aufgrund der identischen Aminosäuresequenzen 3-7 und 9-13 eine entscheidende Rolle. Aus diesem Grunde werden für die Aminosäuren 9 und 10 im Gegensatz zu vorher (Aminosäuren 3,4) die Kodons TTT und ATT bevorzugt.

Erläuterung der Tabelle

Die als Anhang F beigefügte Tabelle ermöglicht es, den Ablauf des Algorithmus Schritt für Schritt „von Hand" nachzuvollziehen. Für jede Startposition werden dabei detailliert alle von der Software gebildeten Kombinations-DNA-Sequenzen (KDS) aufgelistet.
Zu jeder möglichen KDS werden folgende Angaben gemacht:

- Die aus der jeweiligen KDS und der bereits optimierten DNA-Sequenz gebildete Testsequenz, welche zur Evaluierung der KDS herangezogen wird

- Die Scores, welche für Codon Usage, GC-Gehalt, Repetitivität und gefundene DNA-Motive („Sites") ermittelt wurden (CU, GC, Rep, Site)
- Das für die jeweilige Testsequenz ermittelte repetitive Element mit dem höchsten Alignment-Score
- Der ermittelte Gesamtscore

Die KDS sind dabei nach fallendem Gesamtscore sortiert, d.h. das erste Kodon der ersten gezeigten KDS wird an die bereits optimierte DNA-Sequenz angefügt.

B.2.4 Syntheseunterstützung

B.2.4.1 Unterteilung in Subfragmente

Da die Länge doppelsträngiger DNA, die direkt aus Oligonucleotiden aufgebaut werden kann, limitiert ist (vgl. Kap. XXX) muss die Gesamtsequenz zunächst in Subfragmente geeigneter Länge aufgespalten werden. Die Subfragmente müssen dabei so generiert werden, dass sie gegenseitig um 4 Nucleotide überlappen. Diese Überlapps bilden später beim Zusammenligieren der Subfragmente die Hybridisierungsstrecken. Aus diesem Grund dürfen keine zwei Überlappsequenzen identisch bzw. invers-komplementär zueinander sein.
Um die optimalen Schnittpositionen zu bestimmen, berechnet GeneOptimizer in einem ersten Schritt, wie viele Subfragmente notwendig sind, um eine vom Anwender vorgegebene maximale Fragmentlänge nicht zu überschreiten. Ausgehend von den mit Hilfe dieser Information berechneten mittleren Schnittpositionen werden diese nucleotidweise in der Sequenz nach rechts verschoben, bis eine Überlappsequenz gefunden wird, deren Alignment mit einer bereits für eine vorhergehende Schnittposition verwendeten Sequenz (oder derem invers-Komplementärem) nicht mehr als zwei Basenübereinstimmungen ergibt. Darüber hinaus darf sie nicht mit einer der vom Anwender vorgegebenen nicht zu verwendenden Sequenzen übereinstimmen. Die entsprechenden Schnittpositionen werden dem Anwender vorgeschlagen und nach Bestätigung werden die Subfragmentsequenzen in die Datenbank abgelegt. In einem weiteren Schritt kann der Anwender nun an die Enden der Subfragmente Linkersequenzen, welche in der Regel Erkennungssequenzen für Restriktionsenzyme enthalten, anfügen.

B.2.4.2 Aufspaltung in Oligonucleotide

Sind die Subfragmente festgelegt, müssen die Sequenzen der zum Aufbau mittels der Ligationsmethode benötigten Oligonucleotide ermittelt werden. Dazu wird zunächst berechnet, in wie viele Ligationsoligonucleotide der Sense-Strang zerlegt

werden muss, um eine vom Anwender vorgegebene maximale Länge eines Oligos nicht zu überschreiten. Aus der Anzahl kann dann die durchschnittliche Oligolänge und damit die Positionen der Schnittstellen in der Sense-Sequenz ermittelt werden. Bei der Festlegung der Fängeroligonucleotide muss beachtet werden, dass im Sinne einer optimalen Hybridisierungsspezifität die Schmelztemperatur aller Ligationsoligo-Fängeroligo-Hybridisierungsstrecken möglichst identisch sein sollte. Zudem muss diese einer vorgegebenen Temperatur, welche durch die optimalen Reaktionsbedingungen der Ligation definiert wird, möglichst nahe kommen.

Für die theoretische Berechnung von Schmelztemperaturen existieren eine Reihe von Näherungsverfahren [Sambrook 2001]. Allgemein bekannt ist z.B. die sogenannte „Wallace Rule", nach der sich die Schmelztemperatur (in °C) als

F. B.2.4.2-1
$$T_m = 2(A+T) + 4(G+C)$$

berechnet, wobei die Basensymbole für die Anzahl der entsprechenden Nucleotide stehen. Für niedrige Ionenstärken besser geeignet ist die ursprünglich von Bolton und McCarthy abgeleitete Formel

F. B.2.4.2-2
$$T_m = 51.5°C + 16.6 * (\log_{10}[Na^+]) + 0.41(\%[G+C]) - \frac{675}{n} - m$$

mit *n* als Anzahl der Basenpaare und *m* als prozentualer Anteil der Basenfehlpaarungen.
Allen einfachen Näherungsformeln gemeinsam ist, dass die Schmelztemperatur auf Grundlage der Basenzusammensetzung (also GC-Gehalt) errechnet wird und eine Sequenzabhängigkeit der Schmelztemperatur vernachlässigt wird.
Den tatsächlichen Werten am nächsten kommen daher die mittels experimentell gewonnener thermodynamischer Daten berechneten Schmelztemperaturen. Bei dem sogenannten „Nearest-Neighbour"-Verfahren geht man davon aus, dass die Sequenzabhängigkeit der thermodynamischen Eigenschaften eines DNA-Duplex sich auf die Interaktionen der in der Sequenz direkt benachbarten Nucleotide beschränkt. Die thermodynamischen Daten eines Duplex können also durch die Summation der Enthalpie und Entropiewerte von überlappenden Zweierblöcken errechnet werden [Breslauer 1986]. Für die 4 Basen der DNA sind genau 10 solcher Blöcke möglich (AA/TT, AT/TA, TA/AT, CA/GT, GT/CA, CT/GA, GA/CT, CG/GC, GC/CG, GG/CC), deren Daten experimentell aus der Aufnahme von Schmelzkurven eines Sets von Duplexen bekannter Sequenz gewonnen werden.

```
5′-GCATTA-3′
3′-CGTAAT-5′
```

Abb. B.2.4.2-1 DNA-Duplex-Beispiel zum Nearest Neighbour-Verfahren

Die ΔH_{NN} und ΔS_{NN} - Werte des in Abb. B.2.4.2-1 gezeigten DNA-Duplexes können also wie folgt aus den tabellierten Δs_i und Δh_i Daten [SantaLucia 1996] berechnet werden:

F. B.2.4.2-3,4

$$\Delta H_{NN} = \Delta h_{GC} + \Delta h_{CA} + \Delta h_{AT} + \Delta h_{TT} + \Delta h_{TA}$$

$$\Delta S_{NN} = \Delta s_{GC} + \Delta s_{CA} + \Delta s_{AT} + \Delta s_{TT} + \Delta s_{TA}$$

Die Schmelztemperatur ist definiert als die Temperatur, bei der die Konzentrationen der beiden komplementären Oligonucleotide der Konzentration des Duplex equivalent sind.
Im Gleichgewicht gilt also für die Gleichgewichtskonstante der Komplexbildung

F. B.2.4.2-5

$$K = \frac{4}{C_{Oligo}}$$

wobei C_{Oligo} die Gesamtoligokonzentration bezeichnet [Borer 1974].
Aus den bekannten Beziehungen F. B.2.4.2-6,7 lässt sich nun eine einfache Formel für die Schmelztemperatur ableiten, wobei zusätzlich ΔS_{init} als Helixinitiationsentropie eingeführt wird.

F. B.2.4.2-6,7

$$\Delta G = \Delta H - T\Delta S$$

$$\Delta G = -RT \ln K$$

F. B.2.4.2-8

$$T_m = \frac{\Delta H_{NN}}{\Delta S_{NN} + \Delta S_{init} - R \ln K}$$

Da die berechneten Schmelztemperaturen für 1-molare NaCl-Lösungen gelten, muss ggf. noch ein empirischer Korrekturterm für die Salzkonzentration eingeführt werden (z.B. + 12.5 log[Na^+]). Experimentellen Daten zufolge scheinen die für 1M NaCl-Lösungen berechneten Werte jedoch auch die Verhältnisse in 0.15 M NaCl mit 10 mM $MgCl_2$ gut wiederzugeben, einer Lösung, die den Verhältnissen in üblichen PCR- und

Ligationspuffern entspricht. Um wirklich exakte Werte zu erhalten, müssten die notwendigen Parameter für die Konzentrationskorrektur jedoch mit den tatsächlich verwendeten Puffern empirisch ermittelt werden. Da in der Praxis bei den Ligationsreaktionen ohnehin mit Temperaturgradienten gearbeitet wird, ist eine exakte Berechnung der absoluten Schmelztemperaturen ohnehin nicht erforderlich; entscheidend ist vielmehr, dass die Schmelztemperaturen der Hybridisierungsstrecken zueinander möglichst identisch sind.

Wie bereits ausgeführt, werden die Bruchstellen in der Sense-Sequenz durch die vom Anwender vorgegebene maximale Oligolänge definiert. Jeweils ausgehend von zwei benachbarten Schnittstellen wird zunächst die Länge des rechten bzw. linken „Armes" der benachbarten Fängeroligos so festgelegt, dass die Schmelztemperatur der Hybridisierungsstrecken mit dem zwischen den beiden Bruchstellen liegenden Ligationsoligo der vorgegebenen Ligationstemperatur möglichst nahe kommt. Bei AT-reichen Sequenzen kann es jedoch vorkommen, dass sich die beiden derart festgelegten Arme überlappen. Daher wird in diesem Fall die zur Errechnung der Armlängen verwendete Schmelztemperatur schrittweise soweit abgesenkt, bis die Überlappung der Hybridisierungstrecken vermieden wird.

B.2.4.3 Überprüfung auf mögliche Probleme bei der Batch-Synthese

Der Aufbau der Subfragmente im Batchverfahren birgt naturgemäß die Gefahr, dass Oligonucleotide mit anderen als den vorgesehenen komplementären Oligonucleotiden hybridisieren. Zwar wird es hierbei nur selten zu einer Ligation nicht zusammengehöriger Ligationsoligos kommen, da diese Reaktion im Prinzip drei „passende" Partner voraussetzt, jedoch stehen die durch Fehlhybridisierungen gebundenen Oligonucleotide der korrekten Ligationsreaktion nicht mehr zur Verfügung. Analoges gilt für die Ausbildung einer Haarnadelschleife innerhalb eines Oligonucleotids.
Idealerweise wurden diese Gefahren bereits bei der Optimierung der Sequenz durch eine Eliminierung von Repetitionen bzw. invers komplementären Repetitionen vermieden. Dennoch soll die Software den Anwender vor möglichen Problemen während der Ligationsreaktion warnen.
Dazu wird ein lokales Alignment für jede mögliche Kombination der in der Reaktionsmischung vorliegenden Oligonucleotide errechnet und der Anwender gewarnt, falls der Score eines Alignments einen vorgegebenen Wert übersteigt. Auf diese Weise kann auch die potentielle Ausbildung von Haarnadelschleifen erkannt werden, da diese die Folge einer symmetrisch-inversen Komplementarität eines Oligonucleotides mit sich selbst ist.

Ebenso wird eine Warnung generiert, falls eine Hybridisierungsstrecke Ligationsoligo/Fängeroligo die erwünschte Schmelztemperatur nicht erreicht.

B.2.5 Analyse der Klonsequenzen

Wie bereits in Kap. A.2 erwähnt, sind die mittels Festphasensynthese hergestellten Oligonucleotide immer zu einem gewissen Prozentsatz mit Fehlern behaftet. Dies können sowohl Deletionen und Insertionen als auch Basenmodifikationen sein. Letztere werden durch die DNA-Polymerase meist nicht als die ursprüngliche Base erkannt und führen so zu Substitutionen. Daher müssen oft die Plasmidinserts von mehreren Dutzend Kolonien sequenziert und mit der zu synthetisierenden Zielsequenz verglichen werden, um ein völlig korrektes Gen in klonaler Reinheit zu erhalten. Es ist offensichtlich, dass das Vergleichen der Sequenzen in einer Hochdurchsatz-Umgebung nicht mehr manuell vorgenommen werden kann.
Da die vom Basecaller des Sequenzers gelieferten Sequenzdaten [Tibbetts 1995] in der Regel einen Sequenzbereich umfassen, welcher mehrere Dutzend Basen vor dem Start des Inserts beginnt und evtl. sogar noch weiter über das Ende der Templatesequenz hinausreicht, bietet sich die Generierung eines „End-Space-Free"-Alignments an. Da die Sequenzierung sowohl mit einem Forward- als auch einem Backward-Primer erfolgen kann, überprüft die Software zunächst, ob durch das Alignment Klonsequenz / Zielsequenz oder Klonsequenz / invers-komplementäre Zielsequenz ein höherer Alignmentscore erreicht werden kann und generiert dann das geeignete Alignment. Im Falle einer perfekten Sequenzierung könnten nun die in den Klonsequenzen enthaltenen Fehler einfach durch Analyse des Alignments erkannt werden. Leider ist auch die Sequenzierung bzw. die Auswertung des Elektropherogramms durch den Basecaller mit Unsicherheiten bzw. Fehlinterpretationen behaftet. Daher werden mehrdeutige Peaks vom Basecaller als „N"s in den Sequenzdaten aufgeführt. Dies gilt ebenso für relativ kleine Peaks, bei denen nicht sicher ist, ob an der entsprechenden Stelle in der Sequenz tatsächlich ein Nucleotid vorhanden ist. Jedoch ist die Wahrscheinlichkeit, dass einem N tatsächlich ein fehlerhaftes Nucleotid in der Klonsequenz entspricht, sehr gering („unwahrscheinlicher Fehler"). Umgekehrt kann im Alignment eine Substitution bzw. Deletion, welche sich an einer im Elektropherogramm eindeutigen Stelle befindet, als sicherer Fehler in der Klonsequenz gewertet werden („sicherer Fehler"). Um ein korrektes Alignment zu erhalten, muss also eine Scoringmatrix benutzt werden, welche das Alignment eines N in der Klonsequenz mit einer beliebigen Base in der Zielsequenz höher bewertet als eine Insertion oder Substitution, aber deutlich schlechter als das Alignment zweier identischer Basen.

Idealerweise würde bereits der Basecaller die Zielsequenz bei der Interpretation des Elektropherogramms mit einbeziehen. Dies ist jedoch aufgrund der speziellen Aufgabenstellung in den verfügbaren Basecallern nicht vorgesehen und die Entwicklung eines eigenen Basecallers, welcher auch speziell auf den genutzten Sequencer abgestimmt sein muss, schien unter Aufwand/Nutzen-Aspekten nicht sinnvoll. Dennoch kann das Elektropherogramm auch nachträglich bis zu einem gewissen Grad in die Auswertung miteinbezogen werden. Dies betrifft insbesondere die Interpretation von Insertionen. Hier kommt es leider gehäuft vor, dass ein im Vergleich zur Höhe der umgebenden Peaks sehr kleiner Peak vom Basecaller eindeutig als spezifisches Nucleotid identifiziert wird, obwohl dies nicht der Fall ist. Hier kann also nicht wie bei den „N"-Insertionen allein aus der Basensequenz auf einen „unwahrscheinlichen Fehler" geschlossen werden. Daher überprüft die Software bei jeder Insertion, ob die Höhe des verursachenden Peaks kleiner als ein bestimmter Prozentsatz (z.B. 25%) der Höhe des letzten korrekten identifizierten Peaks dieser Base ist. In diesem Fall wird die Insertion als „unwahrscheinlicher Fehler" klassifiziert.

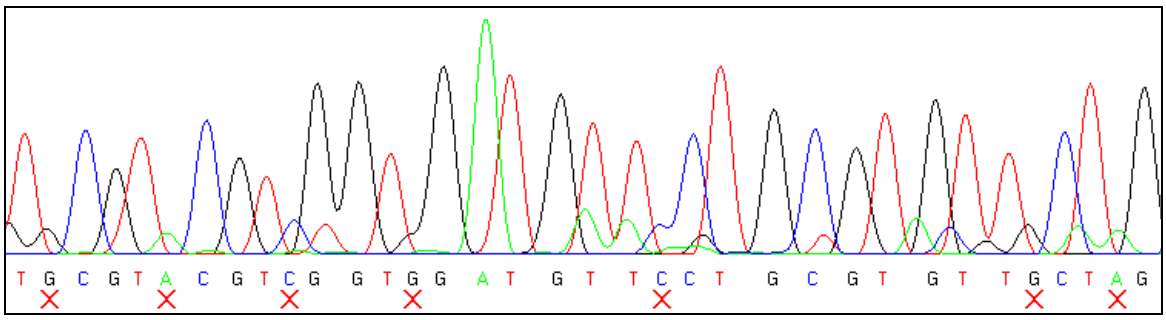


Abb. B.2.5-1 Beispiel einer „schlechten" Sequenzierung mit vielen vom Basecaller fehlerhaft eingefügten Nucleotiden (durch ein rotes Kreuz gekennzeichnet)

Die Bewertung einer Klonsequenz wird nun dergestalt vorgenommen, dass die Zahl der „unwahrscheinlichen Fehler" $n_{unwahrsch.\ Fehler}$ mit einem bestimmten Faktor $g_{unwahrsch.\ Fehler}$ gewichtet wird, ebenso wird die Zahl der „sicheren Fehler" $n_{sichere\ Fehler}$ mit einem betragsmäßig größeren Faktor $g_{sichere\ \ Fehler}$ gewichtet. Gegebenenfalls kann eine differenziertere Betrachtung nach Art des Fehlers (Insertion, Substitution, etc.) vorgenommen werden.

$$Fehlerscore = n_{\substack{sichere \\ Substitutionen}} * g_{\substack{sichere \\ Substitutionen}} + n_{\substack{sichere \\ Insertionen}} * g_{\substack{sichere \\ Insertionen}} + n_{Deletionen} * g_{Deletionen}$$
$$+ n_{\substack{unwahrsch. \\ Substitutuinen}} * g_{\substack{unwahrsch. \\ Substitutionen}} + n_{\substack{unwahrsch. \\ Insertionen}} * g_{\substack{unwahrsch. \\ Insertionen}}$$

F. B.2.5-1

Anschließend werden die Klonsequenzen nach ihrer Güte sortiert. Vor allem bei schlecht interpretierbaren Elektropherogrammen ist die Notwendigkeit einer abschließenden Beurteilung der Klonsequenz durch den Anwender unvermeidbar. Dazu bietet die Software die Möglichkeit, auf einfache Weise das Elekropherogramm zusammen mit dem Alignment zu sichten. Hierbei ist jedoch die Wahrscheinlichkeit sehr groß, dass die in der automatischen Auswertung als fehlerfrei eingestufte bzw. mit dem höchsten Gütescore bewertete Klonsequenz tatsächlich korrekt ist.

C Ergebnisse - Programmbeschreibung

C.1 Gesamtkonzeption

Grundsätzlich gliedert sich die Softwaresuite in zwei Teile: den GeneOptimizer-Teil, der die Arbeitsschritte vom Design und der Analyse der Sequenz bis hin zur Generierung der Oligonucleotidsequenzen umfasst, und den Sequenzanalyse-Teil zur Evaluierung der Klonsequenzen.

Der GeneOptimizer präsentiert sich dem Anwender mit drei Fenstern. Zentral ist der Sequenzeditor, welcher die Eingabe und Bearbeitung der Gensequenz gestattet. Über PullDown-Menüs kann der Anwender von hier aus die meisten Funktionen der Software aufrufen. Die erforderlichen Parameter lassen sich auf den in Funktionsgruppen untergliederten Reiter-Dateikarten auf dem Eigenschaftenfenster einstellen. Dort finden sich auch Funktionen wie die Subfragment- oder DNA-Motivverwaltung.

Über das Meldungsfenster kommuniziert die Software mit dem Anwender, indem sämtliche während der Programmausführung generierten relevanten Informationen in das Textfeld eingetragen werden. Dieses kann im Bedarfsfall auch über die „Leeren"-Schaltfläche geleert werden.

Gerade in Hinblick auf eine GLP-orientierte Gensynthese ist es unerlässlich, sämtliche vom Anwender durchgeführten Schritte mit Datum, Uhrzeit und Benutzerkennung zu protokollieren. Hierzu gehören z.B. Änderungen an der Sequenz, das Durchführen von Optimierungen, das Erzeugen der Subfragmente und Oligonucleotide etc. Zu diesem Zweck fügt die Software nach der Durchführung eines Bearbeitungsschrittes eine Zeile an die im unteren Teil des Meldungsfensters befindliche Protokolltabelle an. Jedoch kann auch der Anwender selbst durch einen Mausklick in die „Aktion"-Spalte der letzten Zeile der Tabelle einen Notiz-Eintrag vornehmen. Datum, Uhrzeit und Benutzerkennung werden von der Software automatisch eingetragen.

Zur Datenanzeige und Visualisierung bedient sich die GeneOptimizer-Suite vorwiegend entweder der Darstellung im HTML-Format oder benutzt die Seitenvorschau der CrystalReports-Berichterstellungskomponente. Während der Programmausführung erzeugte HTML-Dateien werden zunächst im projektspezifischen Arbeitsverzeichnis abgelegt und anschließend automatisch im Standard-Browser des Anwenders angezeigt. Von hier aus können sie unter Benutzung der Browserfunktionen beispielsweise ausgedruckt werden.

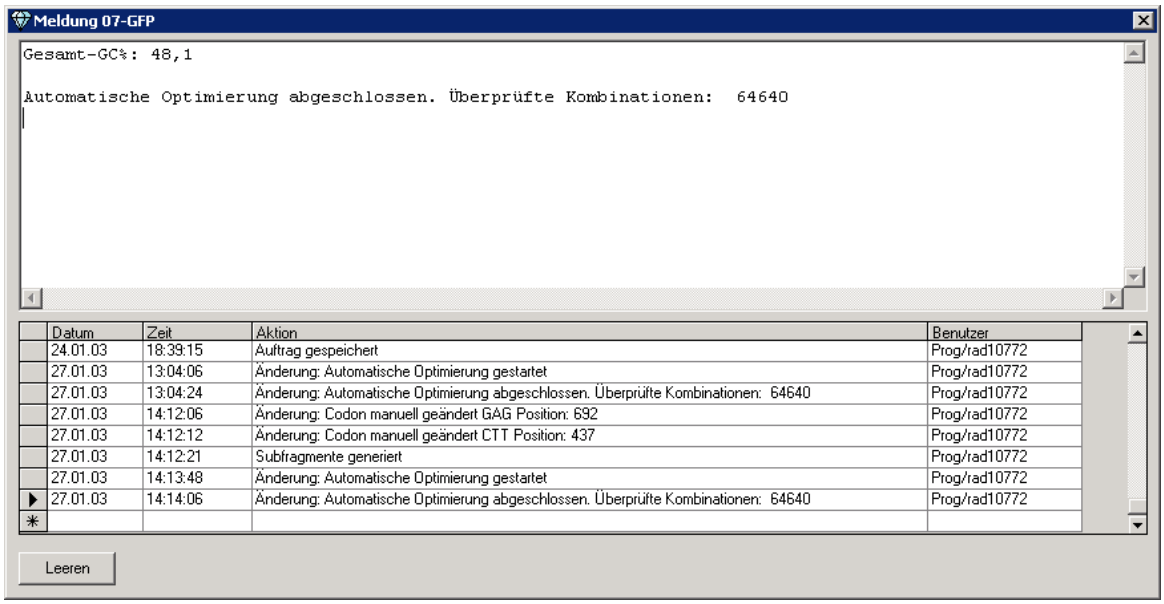


Abb. C.1-1 Meldungs-Fenster

Während das HTML-Format in erster Line zur Darstellung farbcodierter Sequenzen genutzt wird, werden die meisten anderen Datendarstellungen über die CrystalReports-Seitenvorschau präsentiert. Diese ermöglicht eine dem späteren Ausdruck entsprechende Darstellung sowohl graphischer als auch textbasierter Informationen („WYSIWYG").

Über am oberen Fensterrand befindliche Symbolschaltflächen können zahlreiche Funktionen zur Weiterverarbeitung aufgerufen werden. So lässt sich der dargestellte Bericht neben der obligatorischen Ausdrucksmöglichkeit (Druckerschaltfläche) in verschiedenen Datenformaten (RTF, PDF, Word, Excel, HTML) abspeichern oder direkt in die entsprechende Anwendung exportieren (Briefschaltfläche). Eine optimale Darstellung des Berichts kann durch eine variable Zoomfunktion erreicht werden. Schließlich kann in Texten gesucht werden und sowohl Diagramme als auch Textfelder können kopiert und via Windows-Zwischenablage in andere Anwendungen eingefügt werden. Das Eingabefeld „Anmerkung" auf der Graphen-Reiterkarte ermöglicht es dem Anwender, eine kurze Bemerkung einzutragen, welche auf den meisten Ausdrucken ausgegeben wird.

Im Folgenden sollen, soweit möglich angelehnt an den natürlichen Arbeitsablauf und anhand eines konkreten Beispiels, der Rückübersetzung der Aminosäuresequenz des grün-fluoreszierenden Proteins (GFP) aus Aequorea victoria (Genbank Accession Nr. M62654) sämtliche Programmfunktionen detailliert beschrieben werden.

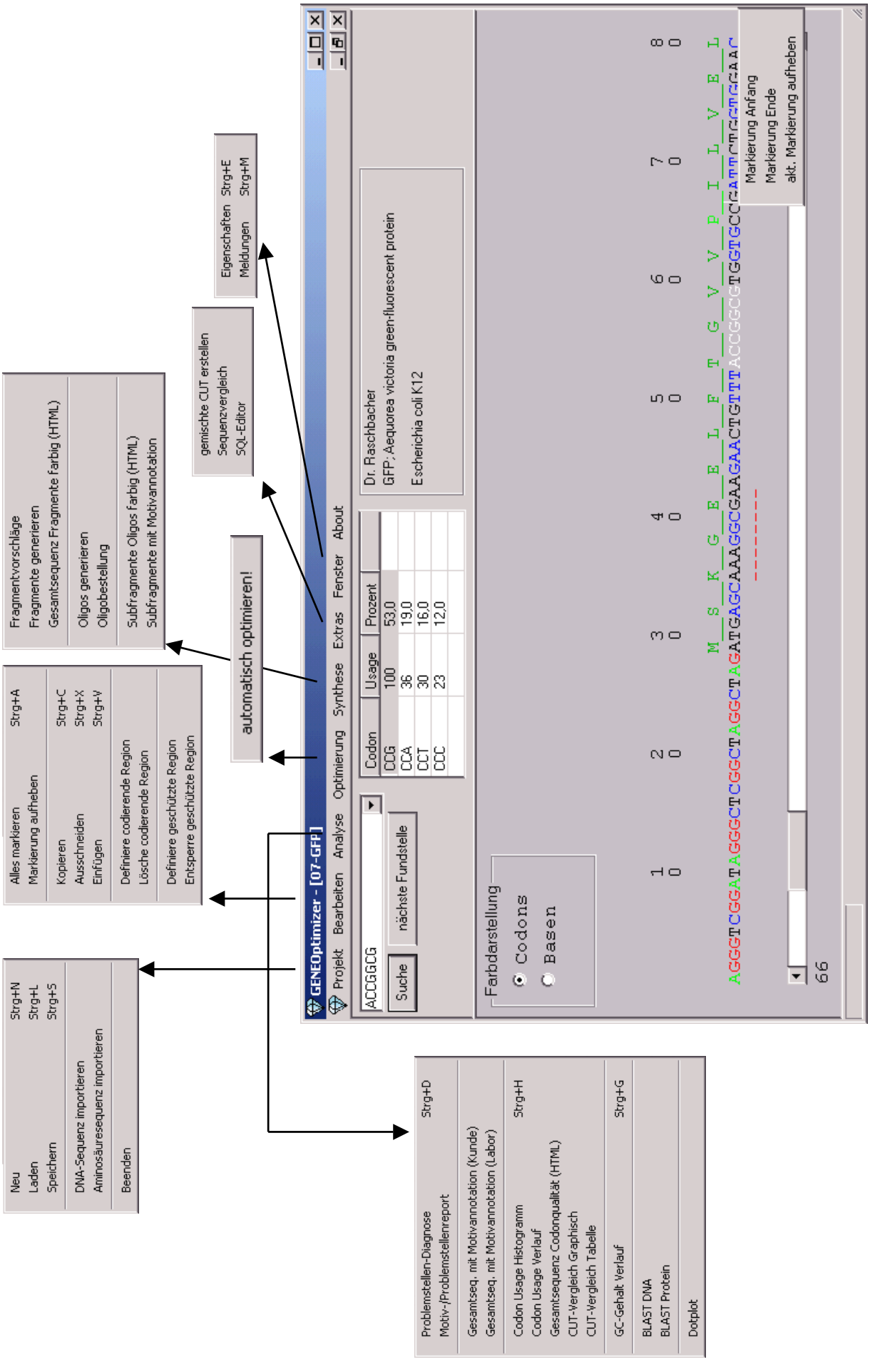


Abb. C.1-2 Der Sequenzeditor im Überblick. Die PullDown-Menüs sind in ausgeklapptem Zustand gezeigt

C.2 Sequenzerfassung und Bearbeitung

C.2.1 Anlegen eines neuen Projekts

Um ein neues Gensyntheseprojekt zu beginnen, muss zunächst über den Menüpunkt „Projekt->Neu" eine neue Projektumgebung, auch als Auftrag bezeichnet, angelegt werden. Dazu kann der Anwender im „Neuer Auftrag"-Fenster eine Bezeichnung vergeben, welche nachfolgend durchgehend zur eindeutigen Identifizierung des Projekts verwendet wird.

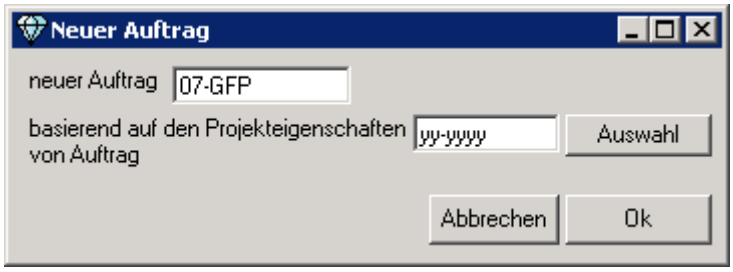


Abb. C.2.1-1 Anlegen einer neuen Projektumgebung

Ein zur rationellen Durchführung vieler Gensynthese-Projekte unerlässliches Hilfsmittel ist die Möglichkeit, mit Template-Aufträgen zu arbeiten. Soweit sinnvoll werden sämtliche für einen gewählten Template-Auftrag vorgenommenen Einstellungen für den neuen Auftrag übernommen. Dies sind z.B. Optimierungsparameter, wie auszuschließende DNA-Motive oder Vorgaben für die Fragment- und Oligogenerierung. Auf diese Weise lassen sich auch eine Reihe von Musteraufträgen für häufig vorkommende Optimierungsprobleme erstellen, wie etwa Optimierung auf Pflanzen, Säuger oder E.coli. Die entsprechende Auftragskennzeichnung muss dazu in das „basierend auf den Projekteigenschaften von"-Textfeld eingegeben werden. Ist diese nicht bekannt, kann über die „Auswahl"-Schaltfläche das Auftragsauswahl-Fenster geöffnet werden.
Die angezeigte Tabelle listet wichtige Datenfelder sämtlicher in der Datenbank gespeicherter Projekte auf, wobei die angezeigten Datenfelder und ggf. Auswahlkriterien durch Modifizierung des angezeigten SQL-Befehls und Ausführen der Abfrage geändert bzw. hinzugefügt werden können. Durch Mausklick auf einen Spaltenkopf der Tabelle kann die Liste nach dem entsprechenden Datenfeld sortiert werden. Durch Anklicken einer Zeile wird die zugehörige Auftragsbezeichnung in das „Auftrag"-Textfeld übernommen und die Auswahl kann durch Betätigung der OK-Schaltfläche bestätigt werden.

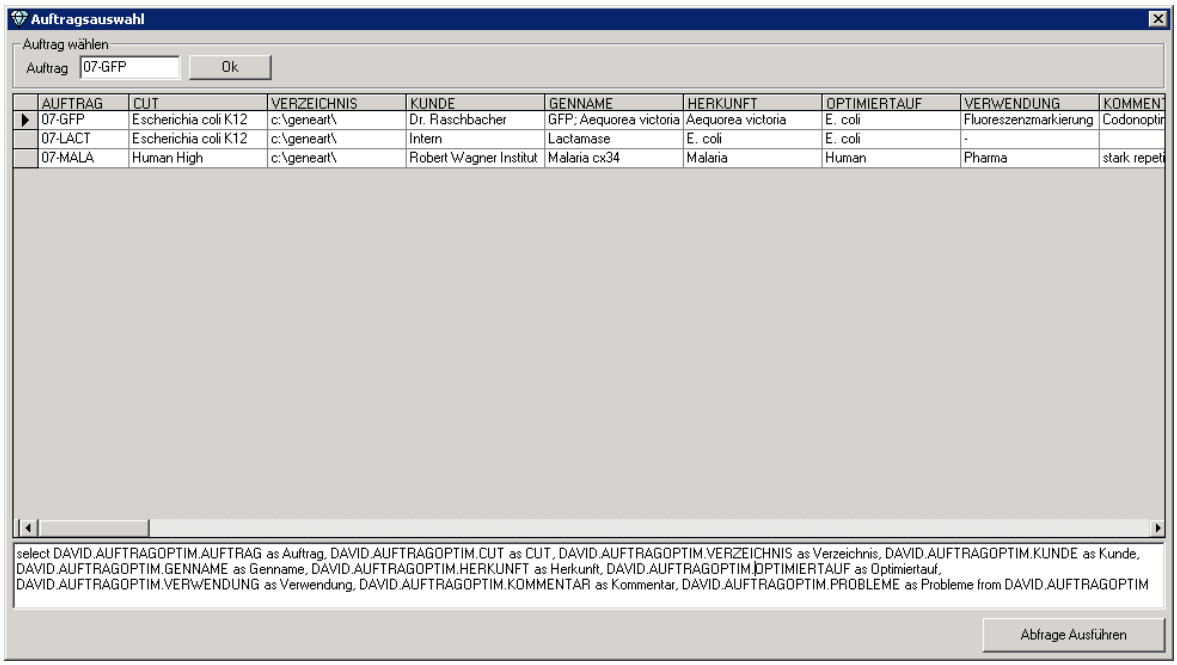


Abb. C.2.1-2 Auswahl eines bereits in der Datenbank erfassten Auftrags

Wird schließlich der „Neuer-Auftrag"-Dialog ebenfalls mit OK bestätigt, öffnet sich ein leeres Sequenzeditorfenster.

Auf der Projekt-Dateikarte im Eigenschaftenfenster können weitere den Auftrag beschreibende Details erfasst werden, wie z.B. die Genbezeichnung oder die geplante Verwendung des synthetischen Gens. Ebenso kann hier das Arbeitsverzeichnis geändert werden. In diesem werden sämtliche während der Projektbearbeitung generierten Dateien abgelegt.

C.2.2 Erfassen und Editieren von Sequenzdaten

GeneOptimizer bietet eine Reihe verschiedener Möglichkeiten, Sequenzinformationen in den Sequenzeditor zu übernehmen. Die rudimentärste Möglichkeit ist natürlich die Erfassung über die Tastatur. Die Software lässt hierbei nur die Eingabe der vier Basensymbole A,T,G und C zu, um Fehleingaben vorzubeugen. Alternativ kann über den Menüpunkt „Bearbeiten->Einfügen" eine DNA-Sequenz aus der Zwischenablage eingefügt werden. Diese wird dabei automatisch gefiltert, so dass z.B. im FASTA-Format vorliegende Sequenzdaten direkt übernommen werden können, da Ziffern, Zeilenumbrüche etc. durch die Software entfernt werden. Die DNA-Seuquenz wird im Sequenzeditor einzeilig dargestellt, wobei die vier Basen mit vier Farben dargestellt werden. Dadurch lassen sich ungewöhnliche Sequenzeigenschaften, wie ein gehäuftes Auftreten eines einzigen Nucleotids leicht erkennen. Die Basenposition wird über der Sequenz in Zehnerschritten angezeigt, wobei die Hunderter-Abstände durch eine Rotfärbung der Ziffern besonders hervorgehoben werden.

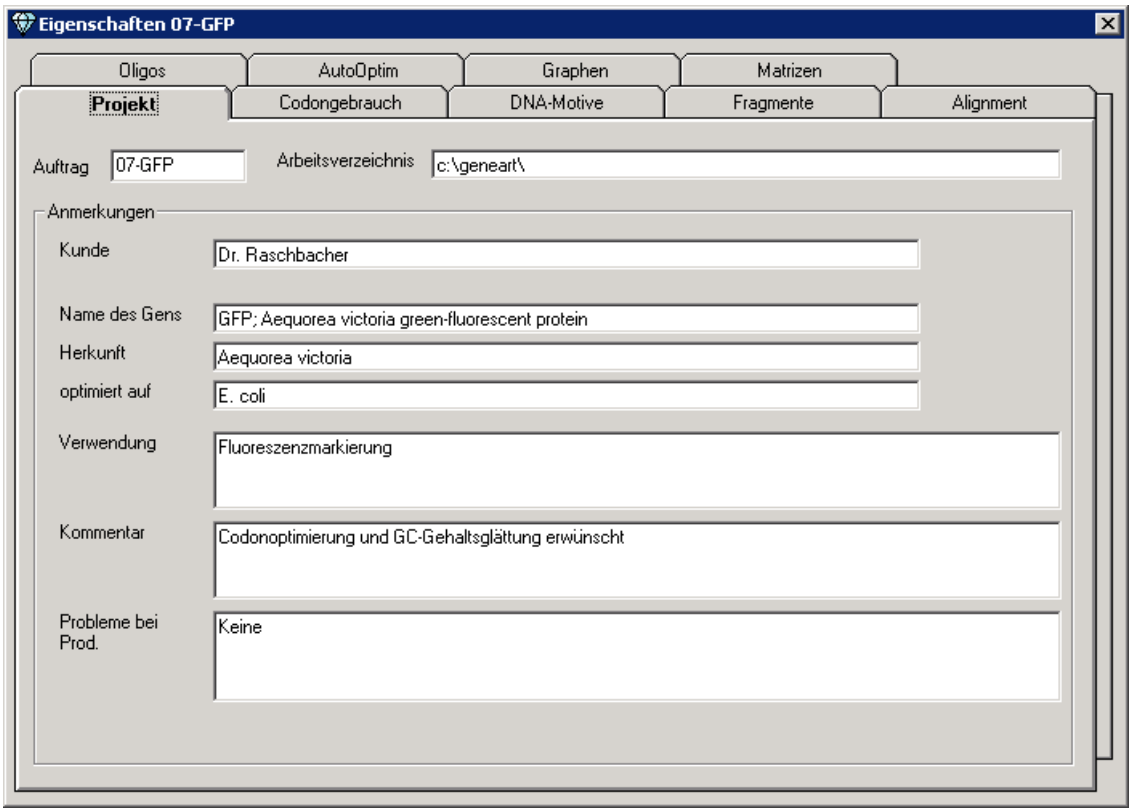


Abb. C.2.2-1 Projekteigenschaftenfenster mit „Projekt"-Reiterkarte im Vordergrund

Über den Schiebebalken unterhalb der Sequenz kann die Eingabemarke basengenau an eine bestimmte Stelle der Sequenz bewegt werden. Dies wird durch die Anzeige der Markenposition am Anfang des Schiebebalkens erleichtert.

Selbstverständlich bietet der GeneOptimizer-Editor alle üblichen Bearbeitungsfunktionen, wie Insertieren oder Löschen von einzelnen Basen. Zum Löschen ganzer Abschnitte muss der entsprechende Bereich zunächst markiert werden. Dazu wird die erste Base des Bereichs mit der rechten Maustaste angeklickt und im aufspringenden PopUp-Menü die Funktion „Markierung Anfang" ausgewählt. Anschließend wird ebenso die letzte Base angeklickt und „Markierung Ende" gewählt. Der ausgewählte Bereich wird nun weiß dargestellt. Über die Menüfunktion „Bearbeiten->Ausschneiden" kann er nun gelöscht werden und die Sequenzinformation ggf. an anderer Stelle oder auch in einem anderen Programm über die entsprechende Menüfunktion wieder eingefügt werden. Analog arbeitet die Menüfunktion „Kopieren", nur dass hierbei der gewählte Bereich nicht gelöscht wird.

An manchen Bereichen muss jedoch gezielt verhindert werden, dass an der Basensequenz Veränderungen vorgenommen werden. Dies kann z.B. der Fall sein, wenn eine Basenabfolge als Erkennungssequenz für ein Protein, wie etwa ein Restriktionsenzym dient. Diese Bereiche können als geschützte Region definiert werden, innerhalb derer sowohl Veränderungen durch den Anwender als auch durch die Software selbst, wie etwa durch eine automatische Optimierung, unterbunden

werden. Dazu wird die betreffende Region zunächst markiert und dann der Befehl „Bearbeiten->Definiere geschützte Region" ausgeführt. Optisch wird der Bereich durch die Unterlegung der betroffenen Basen mit einem roten Querstrich markiert.
Befindet sich die Eingabemarke innerhalb der geschützten Region, kann der Schutz über „Bearbeiten->Entsperre geschützte Region" später auch wieder aufgehoben werden. Damit kann z.B. ein transienter Schutz während einer automatischen Optimierung gesetzt werden, die Kodonwahl aber nach Aufhebung des Schutzes manuell noch verändert werden.
Über die „Speichern"-Funktion des „Projekt"-Menüs können nicht nur die Sequenzdaten, sondern auch alle Projekteigenschaften, wie etwa verwendete Codon-Usage-Tabelle, Optimierungsparameter etc. abgespeichert werden. Soll ein Projekt zu einem späteren Zeitpunkt wieder geladen werden, so kann nach Aufruf der „Laden"-Funktion die Auswahl des Projekts über denselben Auftragsauswahl-Dialog erfolgen wie bei der Neuanlage.

C.2.3 Erfassen und Hinterlegen von Codon-Usage-Tabellen

Wenngleich das GeneOptimizer-System prinzipiell zur Produktionsunterstützung beliebiger DNA-Sequenzen geeignet ist, wird das zu synthetisierende DNA-Fragment in der Regel einen für ein Protein codierenden Bereich enthalten, welcher mit Hilfe der Software für den gegebenen Anwendungszweck optimiert werden soll. Dazu muss zunächst eine Codon-Usage-Tabelle für das gewählte Expressionssystem hinterlegt werden. Die dazu benötigten Daten können beispielsweise von der Kazusa Codon Usage Database auf http://www.kazusa.or.jp/codon/ bezogen werden. Diese Datenbank wird in regelmäßigen Abständen aus den in der NCBI Gene-Bank gespeicherten Sequenzen generiert [Nakamura 2000].
Auf der Kazusa-Homepage kann entweder nach einem bestimmten Organismus gesucht werden oder dieser aus einer alphabetischen Liste ausgewählt werden. Die angezeigten Kodon-bezogenen Nutzungsdaten müssen zur weiteren Verwendung im GeneOptimizer noch mit dem passenden genetischen Code verknüpft werden. Die nach Wahl der entsprechenden Einstellungen generierte Liste im GCG-Stil kann nun im WEB-Browser kopiert und in den GeneOptimizer importiert werden. Dazu muss die kopierte Tabelle auf der Kodongebrauch-Karte im Eigenschaftenfenster in das entsprechende Textfeld eingefügt werden. Nach Eingabe der Organismenbezeichnung kann die Tabelle durch Betätigen der „Importieren"-Schaltfläche in die GeneOptimizer-Datenbank übernommen werden. Bereits in der Datenbank befindliche Codon-Usage-Tabellen können mit Hilfe der Organismus-DropDown-Liste ausgewählt werden. Die Kodonwahl-Daten des selektierten Organismus werden in der

unter der DropDown-Liste befindlichen Tabelle angezeigt und können dort auch bei Bedarf editiert werden.

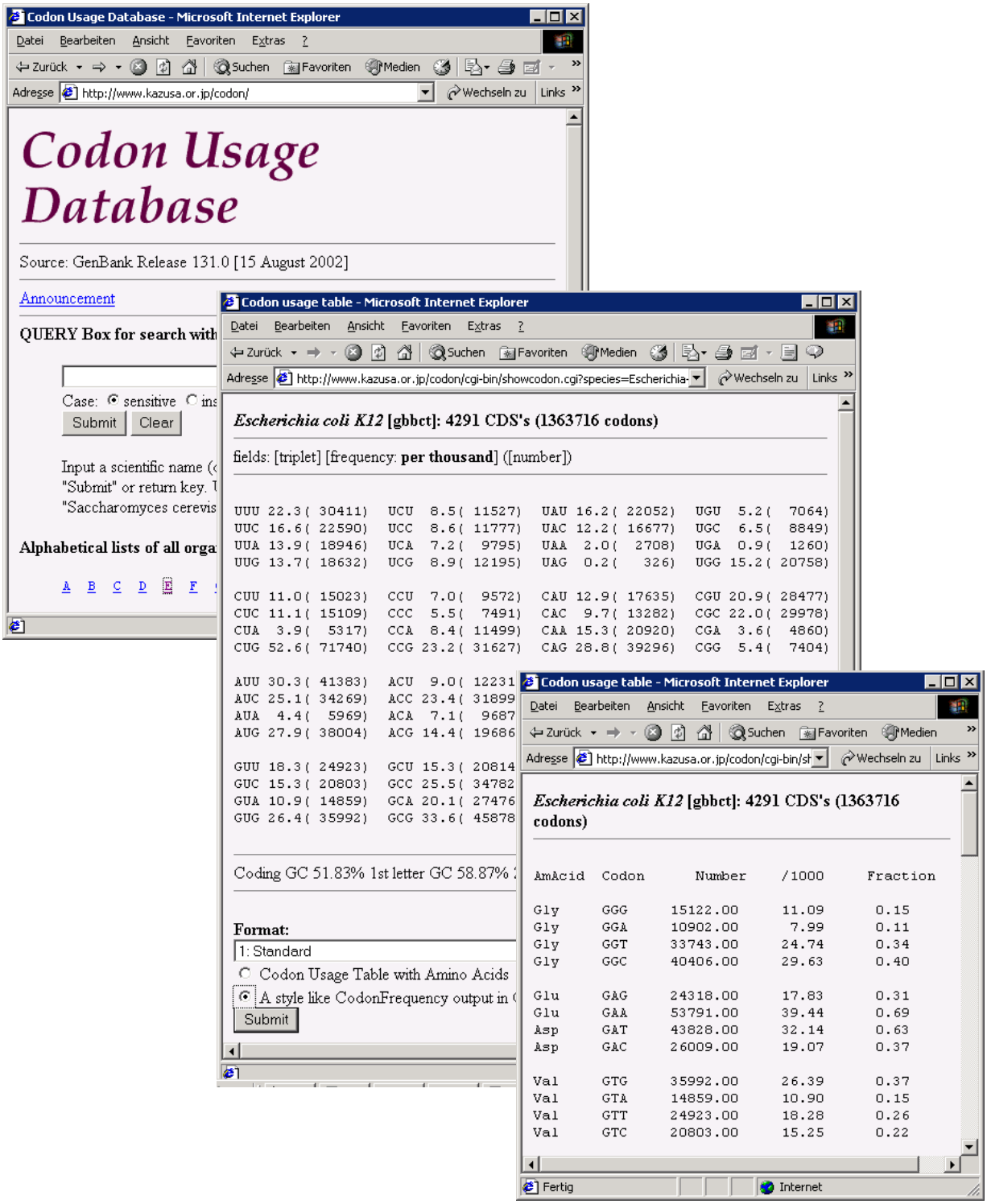


Abb C.2.3-1. Die drei Masken illustrieren, mit welchen Arbeitsschritten man auf der Kazusa-Website Zugriff auf die Codon-Usage-Daten eines bestimmten Organismus erhält. Diese können dann nach Ausgabe im GCG-Datenstil in GeneOptimizer importiert werden.

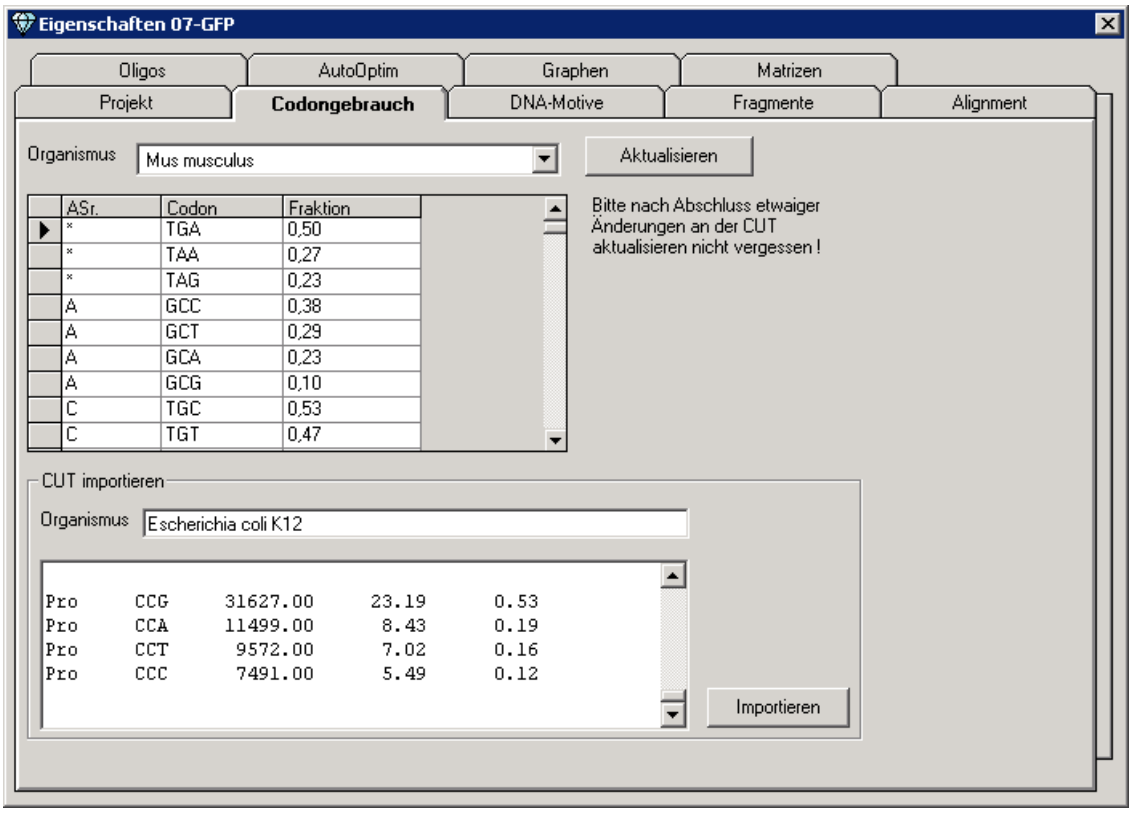


Abb. C.2.3-2 Reiterkarte zur Auswahl und zum Import von Codon-Usage-Tabellen

Soll ein und dasselbe synthetische Gen zur Expression in zwei oder mehreren unterschiedlichen Organismen verwendet werden, so empfiehlt es sich, eine gemischte Codon-Usage-Tabelle zu berechnen. Hierbei wird eine Kompromiss-CUT erstellt, in der Kodons, welche in allen ausgewählten Organismen häufig oder zumindest nicht selten verwendet werden, einen relativ hohen *W*-Wert zugewiesen bekommen, und Kodons, welche in einem der Organismen selten genutzt werden, einen niedrigen *W*-Wert erhalten. Über „Extras->gemischte" CUTS erstellen kann der entsprechende Dialog aufgerufen werden. Hier kann der Anwender in der mit „Ausgangs-CUTs" bezeichneten Liste, die alle bereits in der Datenbank gespeicherten CUT's enthält, zwei oder mehrere Organismen auswählen. In der Praxis ist natürlich eine sinnvolle Optimierung der Kodonwahl des synthetischen Gens nur möglich, wenn die gewählten Organismen eine nicht zu stark unterschiedliche Kodonwahl aufweisen oder aber wenn bei sehr verschiedenen Ursprungs-CUT's die Zahl der Organismen auf zwei oder drei beschränkt wird. Zudem kann es sinnvoll sein, Kodons, welche in *einer* der Ausgangs-CUT's einen W-Wert unter einem gegebenen Schwellwert besitzen, der vom Anwender in das „minimale Usage"-Textfeld eingetragen werden kann, für die Optimierung auszuschließen. Diese erhalten in der generierte CUT eine relative Adaptiveness von 0 zugewiesen. Nach Eingabe einer Bezeichnung für die zu berechnende gemischte Ziel-CUT kann diese über die „Berechnen"-Schaltfläche

erstellt werden. Sie ist nun regulär in der Datenbank gespeichert und kann jederzeit über die „Kodonwahl"-Dateikarte ausgewählt werden.

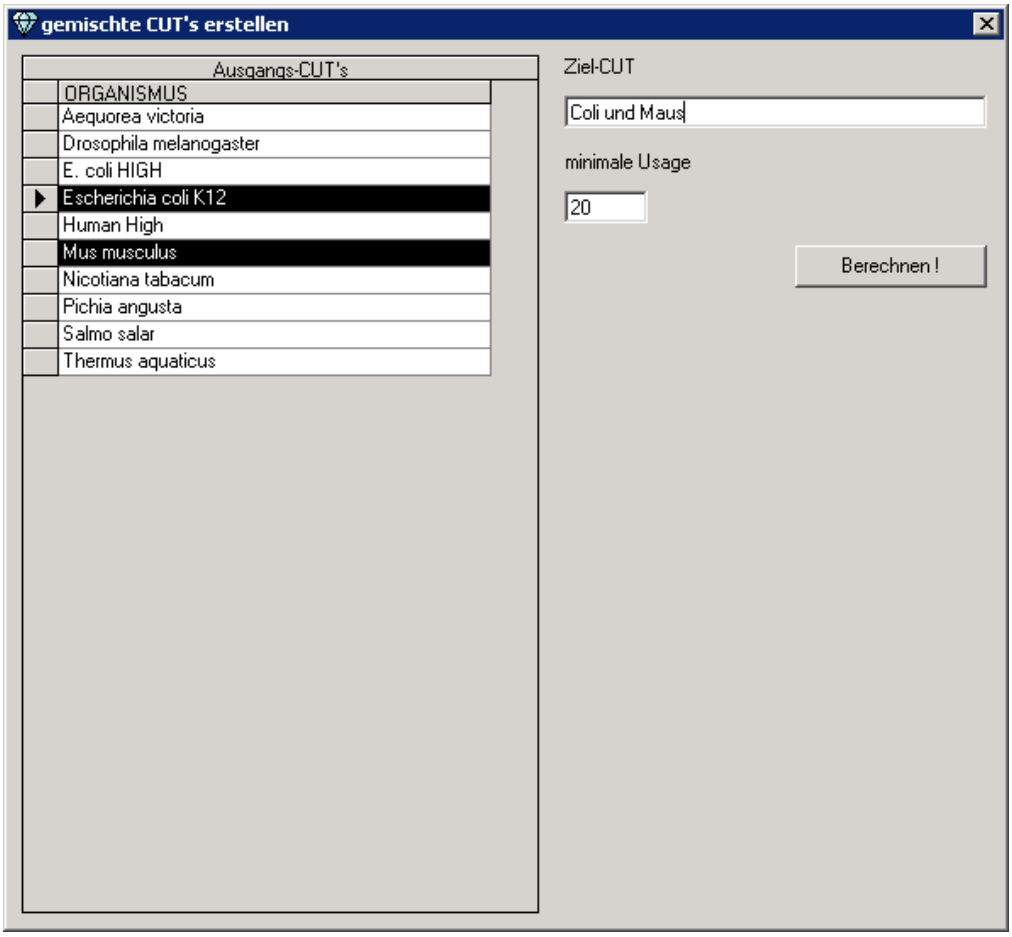


Abb. C.2.3-3 Dialog zur Berechnung gemischter Codon-Usage-Tabellen

C.2.4 Definieren codierender Regionen

In GeneOptimizer kann ein Bereich als codierende Sequenz definiert werden, indem der betreffende Sequenzabschnitt zunächst markiert wird, und anschließend die Menüfunktion „Bearbeiten->Definiere codierende Region" ausgeführt wird. Ein so definierter codierender Bereich kann auch wieder gelöscht werden, indem die Eingabemarke innerhalb dieses Bereiches positioniert wird und die Menüfunktion „Bearbeiten->Lösche codierende Region" gewählt wird.

Sobald eine codierende Region erstellt wurde, werden in diesem Bereich über den Kodons die codierten Aminosäuren im Einbuchstabencode angezeigt, StoppKodons werden durch ein rotes Sternchen gekennzeichnet. Um die Kodons innerhalb codierender Regionen optisch leichter erkennbar zu machen, können in einer alternativen Farbdarstellung aufeinanderfolgende Kodons abwechselnd blau und schwarz dargestellt werden. Die Aminosäure, innerhalb deren Kodon sich die Eingabemarke befindet und welches dadurch selektiert ist, wird hervorgehoben

angezeigt. Zugleich werden die Codon-Usage-Werte aller synonymen Kodons in einer Tabelle in der oberen Hälfte des Editorfensters angezeigt (Die Zeile des aktuell verwendeten Kodons ist dabei grau hinterlegt). Auf diese Weise ist für den Anwender sofort erkennbar, ob an der entsprechenden Stelle ein gutes oder schlechtes Kodon verwendet wird. Durch Wahl eines alternativen Kodons aus der Liste kann der Anwender auf einfache Weise das in der Sequenz verwendete Kodon verändern. Das Einfügen oder Löschen von einzelnen Basen über die Tastatur ist innerhalb codierender Regionen nicht möglich, um z.B. eine ungewollte Verschiebung des Leserasters zu verhindern.
Prinzipiell können in der GeneOptimizer-Software innerhalb einer Sequenz mehrere voneinander unabhängige codierende Bereiche definiert werden. Kann sich eine Programmfunktion nur auf jeweils einen Bereich beziehen (z.B. Codon-Usage Histogramm) , so muss dieser zuvor ausgewählt werden, indem die Eingabemarke innerhalb diese Bereichs positioniert wird.

Ist eine CUT für das aktuelle Optimierungsprojekt ausgewählt, kann auch die Aminosäuresequenz-Importmöglichkeit von GeneOptimizer genutzt werden, welche über „Projekt->Aminosäuresequenz importieren" aufgerufen wird. Nachdem der Anwender die Aminosäuresequenz im Einbuchstaben-Code in das Textfeld des Dialogs eingegeben oder über die „Laden"-Funktion aus einer Datei eingelesen hat und die Eingabe mit OK bestätigt wurde, nimmt die Software automatisch eine Rückübersetzung unter Verwendung der am häufigsten genutzten Kodons vor. Die generierte DNA-Sequenz ist dabei über Ihre ganze Länge als codierende Sequenz definiert, so dass die Aminosäuresequenz unmittelbar nach dem Import über der DNA-Sequenz angezeigt wird.

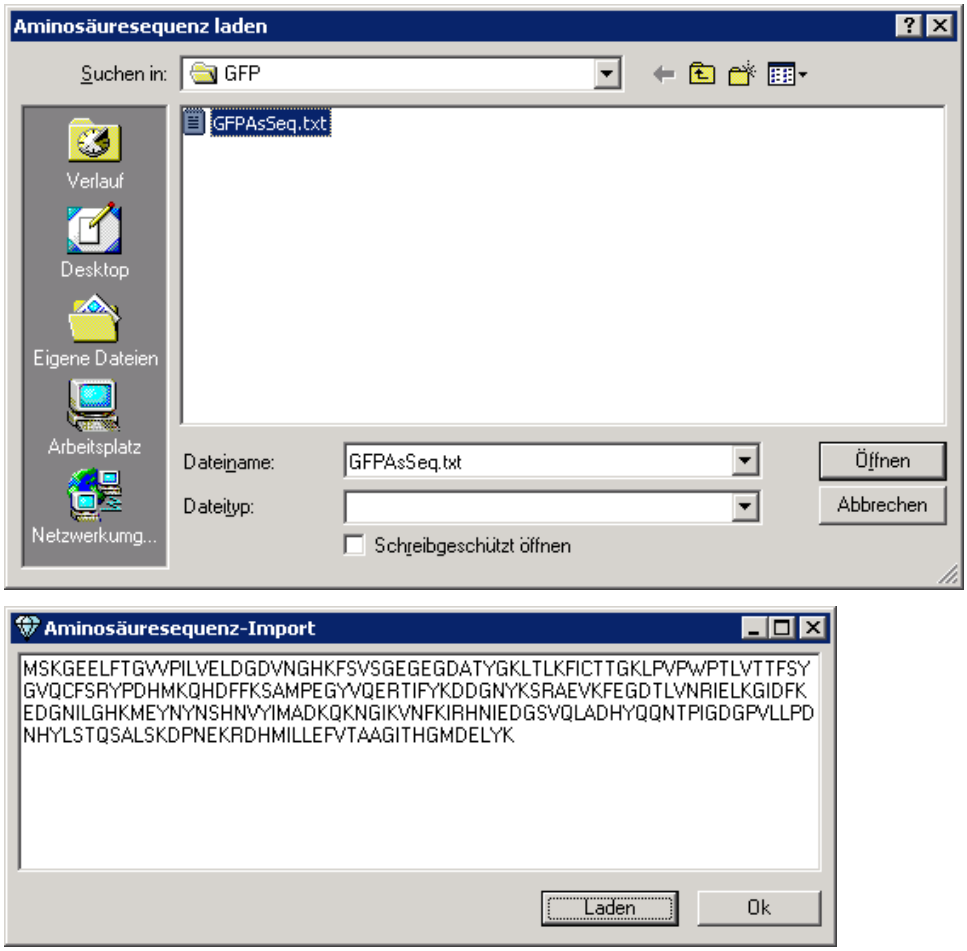


Abb.C.2.4-1 Dialog zum Aminosäuresequenzimport. Über „Laden" kann der Dateiauswahl-Dialog aufgerufen werden. Nach Wahl einer Datei wird diese eingelesen und im Textfeld angezeigt.
Im Beispiel ist dies die Aminosäuresequenz des grün-fluoreszierenden Proteins (GFP) aus Aequorea victoria (Genbank Accession Nr. M62654)

C.3 Sequenzanalyse

Sowohl vor als auch nach der Optimierung ist eine detaillierte Analyse der Sequenz unerlässlich. Sie soll dem Anwender zum einen eine Abschätzung ermöglichen, wie nahe die Sequenz den experimentellen Erfordernissen kommt und ihn zum anderen auf mögliche Probleme während der Herstellung oder auch in der späteren Anwendung hinweisen. Idealerweise führt die Kombination von Analyse und Optimierung zu einem Rückkopplungseffekt, durch den der Anwender die Parameter der Optimierung interaktiv so lange verändert, bis die Merkmale der Sequenz seinen Vorstellungen entsprechen.

C.3.1 DNA-Motivverwaltung

Eine der größten Gefahren einer auschließlich Kodonwahl-orientierten Optimierung, wie sie von vielen Standard-Backtranslation-Programmen vorgenommen wird, besteht in der ungewollten Einführung von DNA-Motiven, welche die Expression mindern oder indirekt toxisch wirken. Dadurch kann die durch die Optimierung der Kodonwahl des synthetischen Gens auf die des Expressionssystems möglicherweise erreichbare Steigerung der Expression vollständig zunichte gemacht werden. Andererseits ist die gezielte und problemlose Einführung bestimmter Motive ein großer Vorteil synthetischer Gene. Daher bietet das GeneOptimizer-System leistungsfähige Möglichkeiten zur Erkennung und datenbankbasierten Verwaltung von DNA-Motiven. Letztere wird auf der Karteikarte „DNA-Motive" im Eigenschaftenfenster vorgenommen.

In der mit „Motive" bezeichneten Tabelle können DNA-Motive vom Anwender erfasst, editiert und ggf. auch wieder gelöscht werden. Als Sequenz können beliebige durch die Regular-Expression-Engine unterstützte Ausdrücke eingegeben werden. In der Regel wird hierbei eine IUPAC-Consensus-Sequenz zur Charakterisierung des Motivs verwendet. Innerhalb dieser sind auch Basenzahl-Angaben, wie etwa in GCTTN(3,5)ATG möglich. Dabei bedeutet N(3,5), dass an der entsprechenden Stelle mindestens 3 und maximal 5 N's (beliebige Basen) vorkommen dürfen. Werden komplexe reguläre Ausdrücke verwendet, bei denen die IUPAC-Basensymbole nicht verwendet werden, so muss dem Ausdruck ein „RE" vorangestellt werden. Mit Hilfe des Ausdrucks „RE (\w{4,})\1\1" können beispielsweise Dreierblöcke von aufeinanderfolgenden identischen Basenabfolgen mit mindestens 4 Basen Länge charakterisiert werden. Die Sequenz AGCCAGCCAGCC etwa würde diesem Ausdruck genügen. Eine letzte Möglichkeit ist, in der Sequenzspalte mittels „Matrix *Name*" auf eine Matrixdefinition zu verweisen, welche auf einer weiteren Karteikarte erfasst

werden kann (s.u.). In der Spalte „Reverse" wird erfasst, ob auch der invers komplementäre Basenstrang durchsucht wird. Im Fall von Consensus-Sequenzen bezeichnet „Gewichtung" den standardmäßig bei der Ermittlung der Gütefunktion für das Auftauchen eines Motivs subtrahierten Wert. Da Matrixähnlichkeiten durch einen Wert zwischen 0 und 100 ausgedrückt werden, bezeichnet die Gewichtung hier den Faktor, mit dem die Matrixähnlichkeit standardmäßig multipliziert wird, bevor sie in die Gütefunktion eingeht.

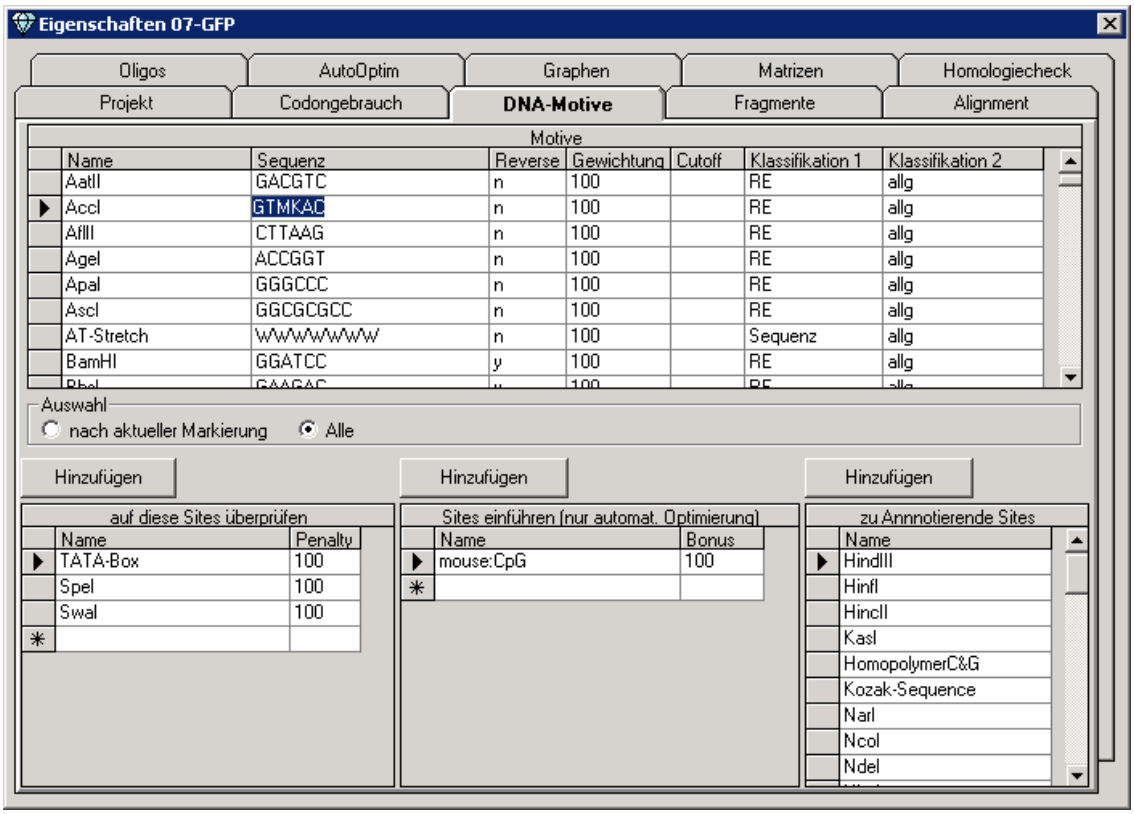


Abb. C.3.1-1 Motivverwaltungs-Reiterkarte

Bei Verwendung von Matrixdefinitionen kann schließlich unter „Cutoff" ein Schwellwert erfasst werden. Liegt der bei Berechnung der Matrixähnlickeit erzielte Wert unterhalb dieses Schwellwertes, wird der entsprechende Bereich innerhalb der DNA-Sequenz nicht als das durch die Matrix spezifizierte Motiv interpretiert.

Um auch bei einer großen Anzahl erfasster Motive die Übersichtlichkeit zu wahren, kann jedes Motiv mit zwei Merkmalen klassifiziert werden. Hierfür bietet sich beispielsweise eine Einteilung nach Funktion (Restriktionsenzyme, Transkriptionsfaktor-Bindungsstellen, ...) und Vorkommen (Eukaryonten, Prokaryonten,...) an. Die Tabelle kann sowohl nach dem Namen der Motive, als auch anhand der Klassifikationsmerkmale sortiert werden, indem der Anwender den Kopf der entsprechenden Spalte anklickt. Weiterhin kann er die angezeigten Motive selektiv einschränken, indem er in einem Feld einen Textabschnitt markiert und den

Radiobutton „Auswahl auf akt. Markierung einschränken" setzt. So kann der Anwender etwa einen Teil einer Sequenz markieren und sich anschließend nur die DNA-Motive anzeigen lassen, deren Sequenz ebenfalls den markierten Bereich enthält. Durch Setzen des „Alle"-Radionbuttons werden wieder sämtliche in der Datenbank erfassten Motive aufgelistet.

Wie bereits erwähnt, können Motive auch durch eine Häufigkeitsmatrix definiert werden. Dies erfolgt auf der Karteikarte „Matrizen". Nach Eingabe einer Matrizenbezeichnung und der Länge erzeugt die Software nach Betätigen der „Anlegen"-Schaltfläche eine bis auf die Positionsangaben leere Tabelle. In diese können für jede Position entweder die tatsächlichen Häufigkeiten der Nucleotide oder die prozentualen Anteile eingegeben werden. Im ersten Fall muss „Fraktionen berechnen" gehakt sein, so dass die Software beim Wechsel in die nächste Zeile die absoluten Häufigkeiten in prozentuale Anteile umrechnet. Der den Informationsgehalt charakterisierende Consensus Index C_i wird ebenfalls beim Zeilenwechsel automatisch berechnet. Bereits in der Datenbank erfasste Matrixdefinitionen können über die Dropdown-Liste geladen und nachfolgend ggf. editiert werden.

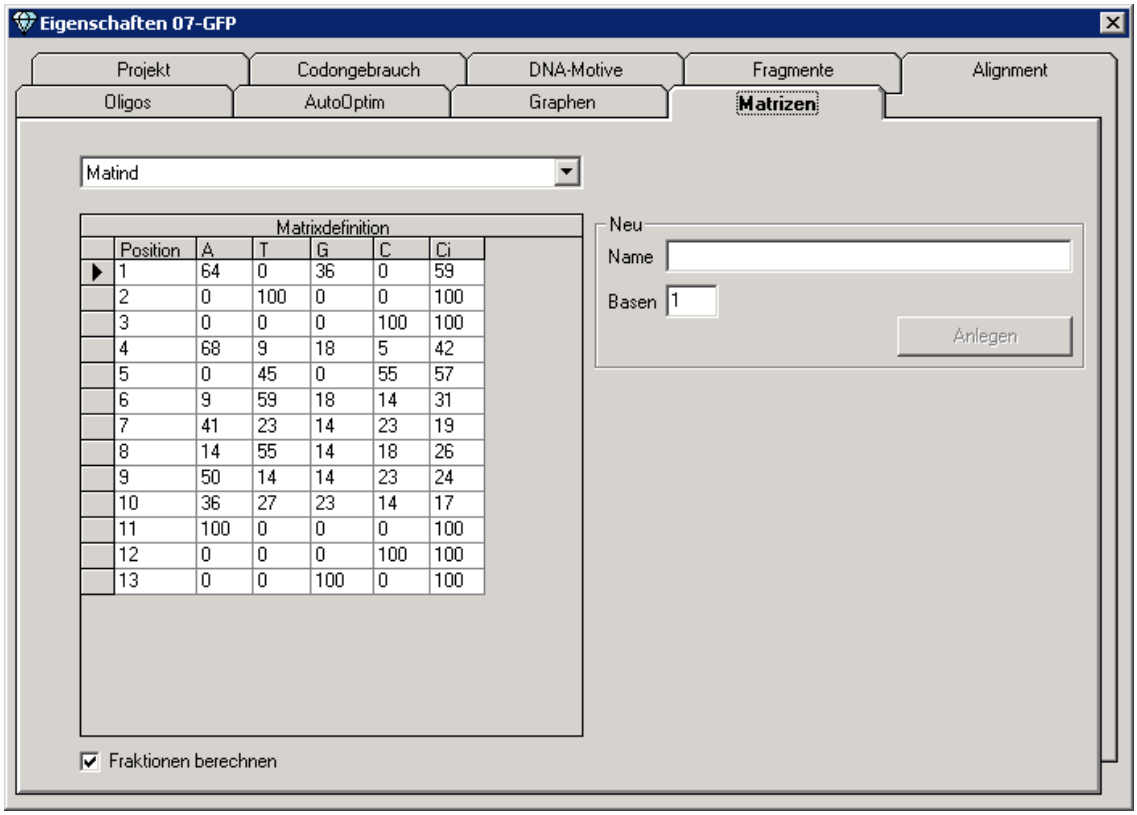


Abb. C.3.1-2 Reiterkarte zum Erfassen von Häufigkeitsmatrizen

C.3.2 Suchen-Funktion des Editors

Eine einfache Möglichkeit, die DNA-Sequenz auf bestimmte Motive hin zu überprüfen, stellt die in den Sequenzeditor integrierte Suchen-Funktion dar. Das für die Suche benutzte Motiv kann als regulärer Ausdruck in das in der linken oberen Ecke lokalisierte Eingabe/DropDownlisten-Kombinationsfeld eingegeben werden. Dabei sind syntaktisch die gleichen Möglichkeiten gegeben, wie unter „Motivverwaltung" beschrieben. Nach Betätigen der „Suche"-Schaltfläche erscheinen alle dem Motiv entsprechenden Sequenzabschnitte weiß markiert. Über „nächste Fundstelle" kann von einer Markierung zur nächsten gesprungen werden. Ist die letzte erreicht, führt eine weitere Betätigung der Schaltfläche zum Anspringen der ersten Fundstelle.
Wird das Kombinationsfeld ausgeklappt, so erscheint alphabetisch geordnet eine Auflistung aller in der DNA-Motivverwaltung erfassten Motive in der Form „*Name* :: *Sequenz*". Wird ein Listenelement ausgewählt, so wird die Sequenz des Motivs in das Eingabefeld übernommen und kann für die Suche benutzt werden.

C.3.3 Problemstellen-Diagnose

Eines der wichtigsten Analysewerkzeuge ist die Problemstellendiagnose, welche über „Analyse->Problemstellendiagnose" aufgerufen werden kann. Sie ermöglicht die Untersuchung der Sequenz sowohl auf das Vorkommen unerwünschter Motive, als auch kritischer Sequenzwiederholungen oder möglicherweise zu Sekundärstrukturen führender invers komplementärer Wiederholungen. Die Berechnung der entsprechenden Alignments erfolgt unter Verwendung der auf der „Alignment"-Dateikarte eingetragenen Parameter (Match-Score, Gap- und Mismatch-Penalty). Dabei werden allerdings für jeden Aufgabentyp (Wiederholungen bzw. invers-komplementäre Wiederholungen) nur die zehn Alignments mit den höchsten Alignmentscores berechnet.
DNA-Motive werden in die Überprüfung miteinbezogen, indem diese in der Motivverwaltungstabelle ausgewählt werden und mit der Schaltfläche „hinzufügen" in die Liste zu überprüfender Motive übernommen werden. Aus dieser Liste können sie natürlich ggf. auch wieder gelöscht werden. Hat der Rechner die Überprüfung der Sequenz beendet, öffnet sich das „Problemstellen-Analyse"-Fenster. Hier findet sich neben der Aufgabenbeschreibung (Überprüfung auf Motive bzw. (invers komplementäre) Wiederholungen) und der Problemnummer eine Beschreibung des Problems sowie Angaben bezüglich der Position und der Länge des problembehafteten Bereichs.

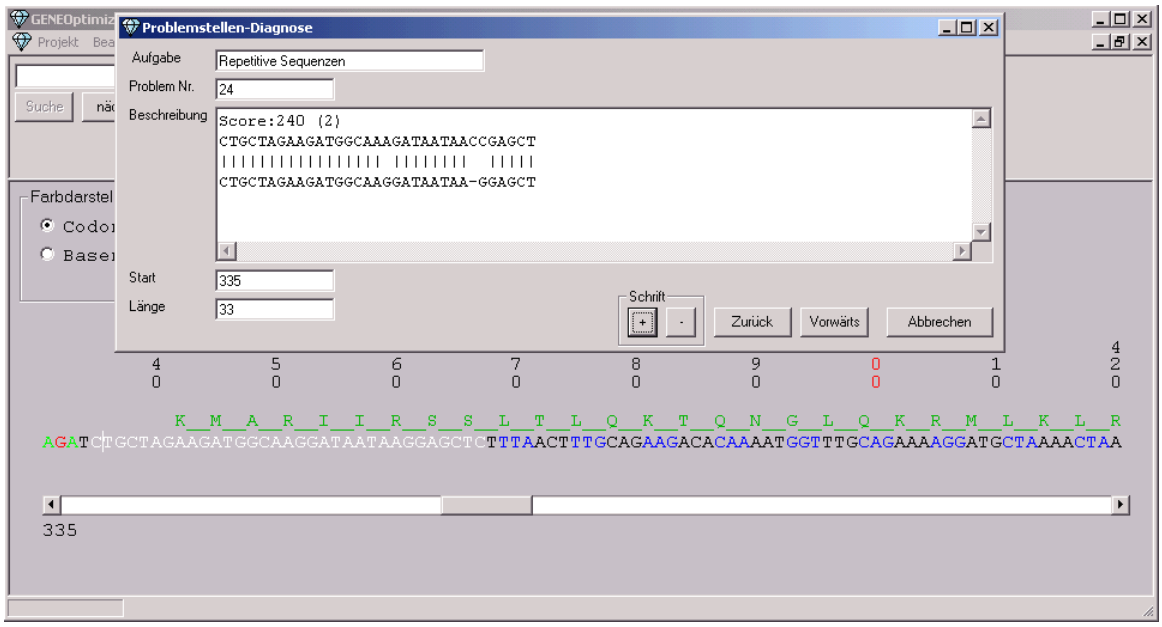


Abb. C.3.3-1 Die im Diagnosefenster gezeigte Problemstelle (ein Part einer Repetition) ist in der Sequenz weiß markiert dargestellt.

Bei der Überprüfung auf Sequenzwiederholungen enthält das Beschreibungsfenster ein Alignment der zueinander ähnlichen Sequenzabschnitte. Um auch längere Alignments übersichtlich darstellen zu können, lässt sich mit den „Schrift +/-„ Schaltflächen die verwendete Schriftart vergrößern bzw. verkleinern. Mit den Weiter/Zurück-Schaltflächen kann der Anwender von einem Problem zum nächsten blättern. Dabei wird gleichzeitig die jeweilige Problemstelle im Sequenzeditor angesprungen und weiß markiert dargestellt. Dies ermöglicht es dem Anwender, die Problemstelle durch eine Veränderung der Sequenz manuell zu entschärfen. So kann z.B. eine unerwünschte Restriktionsschnittstelle innerhalb einer codierenden Sequenz durch die Wahl eines alternativen Kodons eleminiert werden.

C.3.4 Motivreport

Eine übersichtliche Zusammenfassung in Hinblick auf Art und Anzahl der in der Sequenz vorhandenen erwünschten und unerwünschten sowie der ausgeprägtesten (invers komplementären) Sequenzwiederholungen bietet die Funktion „Analyse->Motiv/Problemstellenreport“. In dem generierten Bericht werden nach DNA-Motiven gruppiert die Anzahl der Fundstellen zusammen mit deren detaillierter Auflistung (als Motiv erkannte Sequenz, Position), sowie die Alignments und deren Score dargestellt. Erstellt man einen Motivreport vor und nach einer (automatischen) Optimierung, so lässt sich sehr leicht erkennen, wie viele erwünschte und unerwünschte Motive in die Sequenz eingeführt bzw. eleminiert werden konnten und in welchem Ausmaß ggf. die (invers komplementäre) Repetitivität der Sequenz verringert werden konnte.

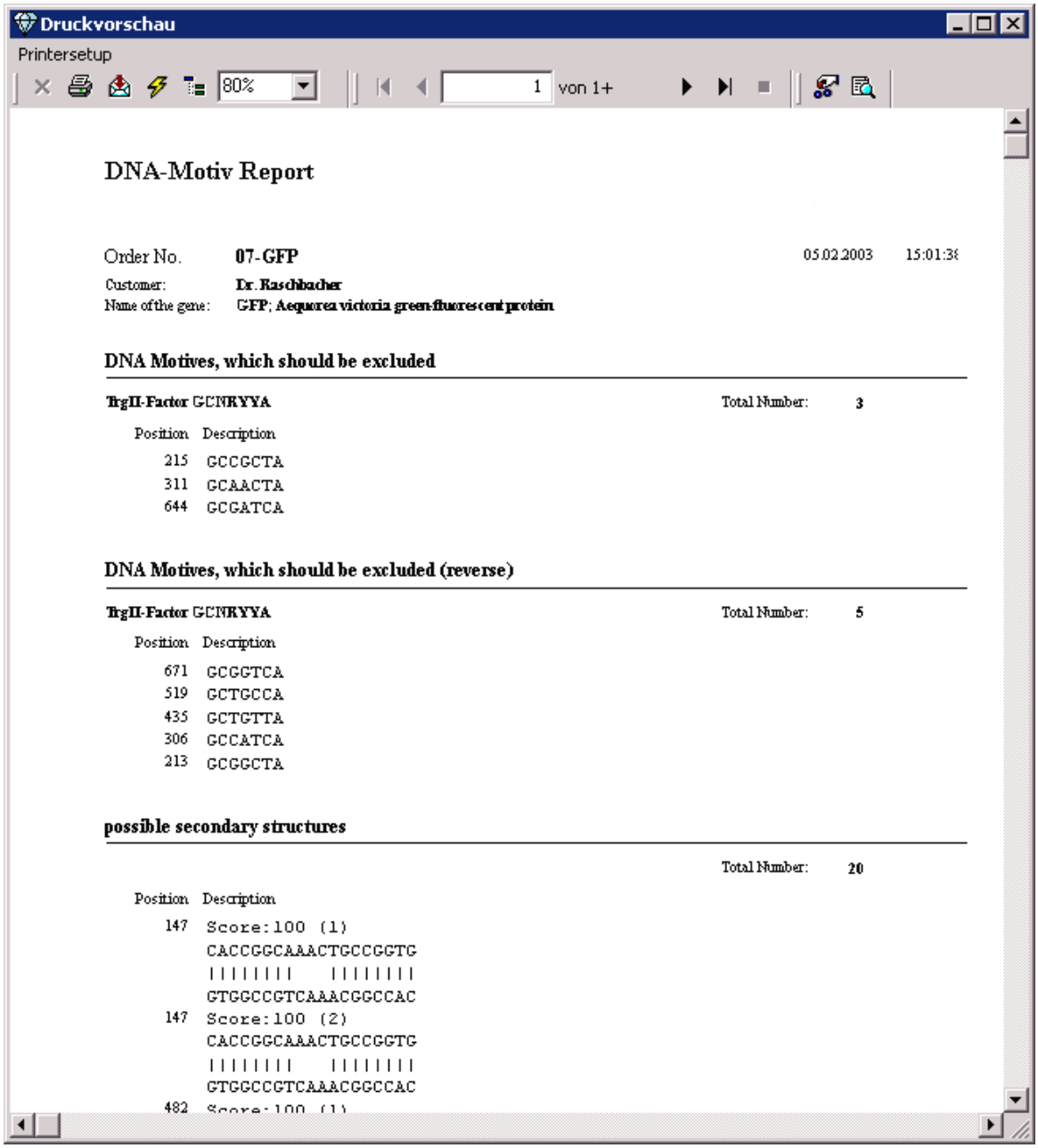


Abb.C.3.4-1 Ausschnitt aus einem Motivreport

C.3.5 Darstellung der Sequenz mit Motivannotation

Ebenfalls zu den Analysewerkzeugen zählt die Darstellung der DNA-Sequenz einschließlich der zugehörigen Aminosäuresequenz mit einer Annotation der vorhandenen DNA-Motive. Deren Name wird an der Stelle des Vorkommens über der Sequenz angezeigt. Der Beginn der Fundstelle ist dabei identisch mit der Position des ersten Buchstabens der Motivbezeichnung.

Um ein Überlappen der Motivbezeichnungen zu verhindern, werden diese ggf. übereinander gestellt gedruckt. Welche Motive in die Annotation miteinbezogen werden, kann der Anwender bestimmen, indem er die gewünschten Motive auf der

Dateikarte „DNA-Motive" aus der Motivtabelle in die Liste „zu annotierende Sites" übernimmt bzw. sie daraus wieder entfernt.

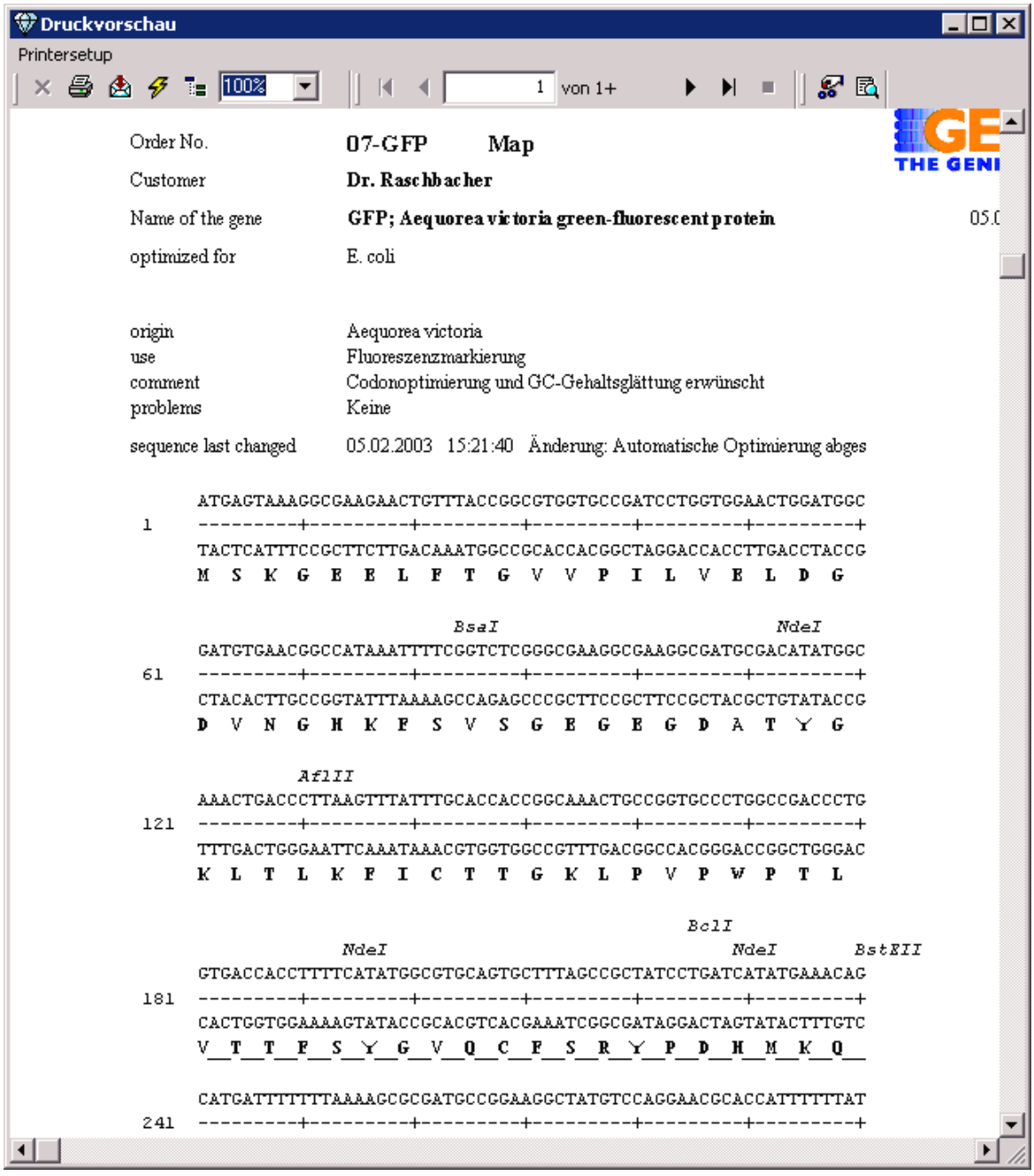


Abb. C.3.5-1 Ausschnitt aus eine Sequenzannotation

C.3.6 Analyse der Kodonwahl

Oftmals ist die Anpassung der Kodonwahl des synthetischen Gens an die des vorgesehenen Expressionssystems ein vordringliches Optimierungsziel. Entsprechend vielfältig sind die angebotenen Analysemöglichkeiten.

Den besten Überblick in Bezug auf die Kodonwahl der codierenden Sequenz bietet das Codon-Usage-Histogramm, welches nach Aufruf der Funktion „Analyse->Codon Usage Histogramm" berechnet wird. Dabei werden die verwendeten Kodons nach ihren *W*-Werten (relative Adaptiveness), mulipliziert mit 100, in 10er Schritten als prozentuale Häufigkeiten histogrammiert. Bei einer hinsichtlich der Kodonwahl optimalen Gensequenz werden sich alle Kodons im „90-100" - Bereich befinden, während sich bei Wildtyp-Genen in Bezug auf die Kodonwahl des gewählten heterologen Expressionssystems oftmals eine breite Verteilung ergibt. Zusätzlich zum Histogramm erhält der Anwender im Meldungsfenster den arithmetischen und geometrischen Durchschnitt der *w*-Werte aller Kodons. Letzterer entspricht dem Codon-Adaption-Index nach Sharp und Li und liefert damit eine zahlenmäßige Aussage darüber, wie gut das Gen der Kodonwahl des Expressionssystems angepasst ist.

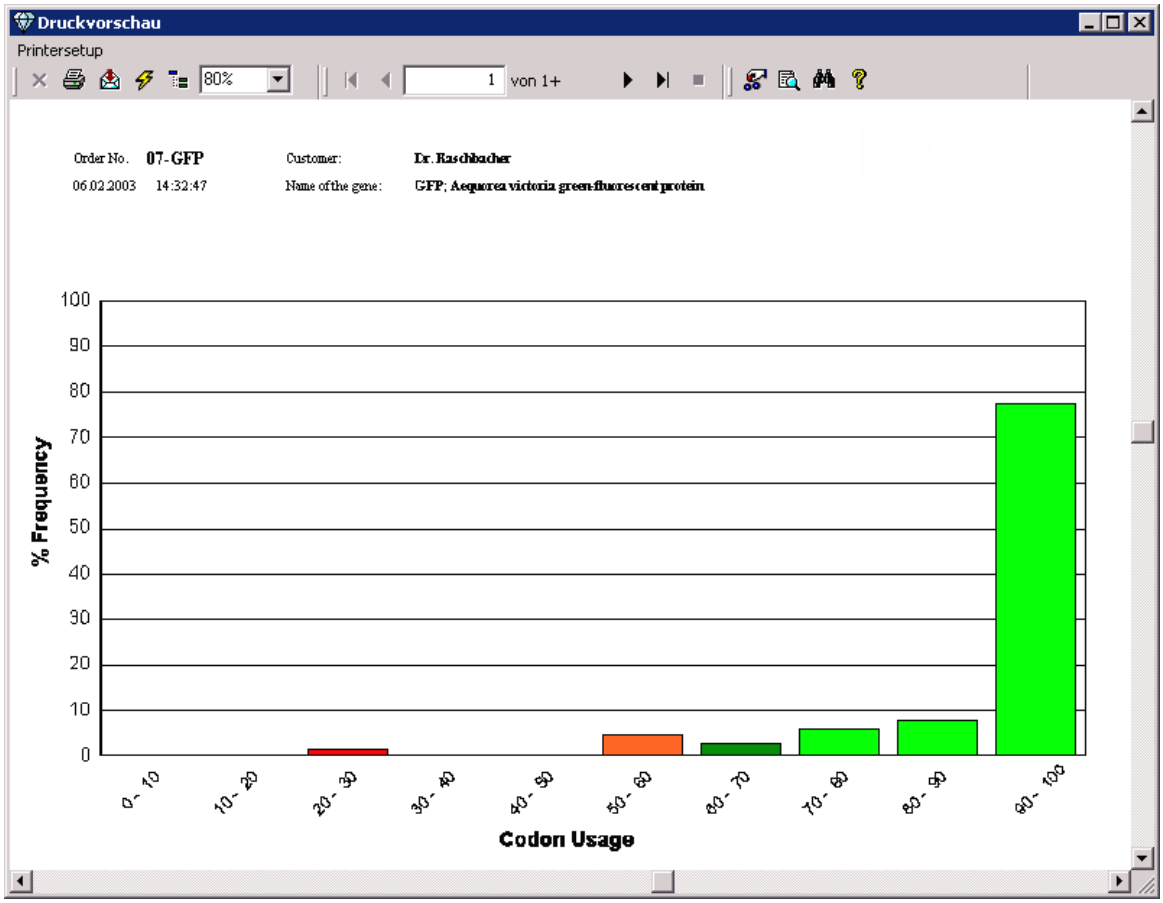


Abb. C.3.6-1 Codon-Usage-Histogramm. Unter „Codon Usage" wird in diesem Fall die „relative Adaptiveness", multipliziert mit 100, eines einzelnen Kodons verstanden

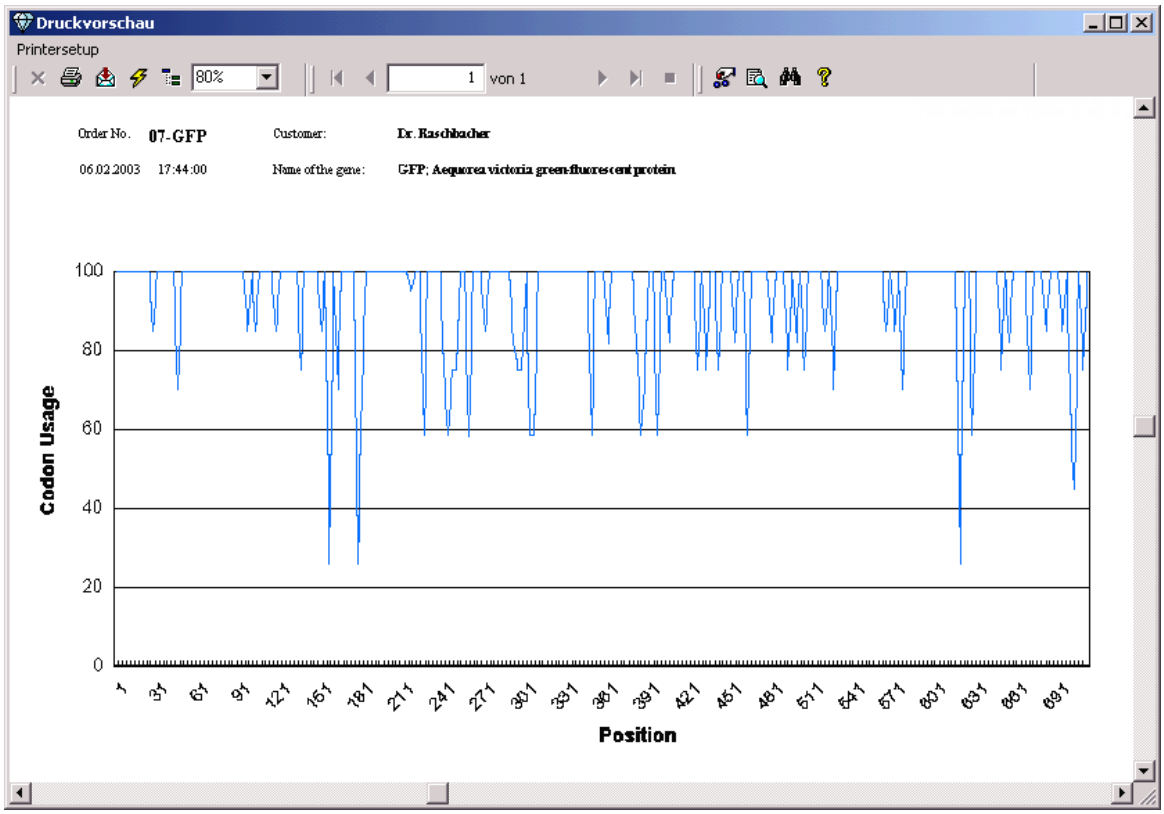
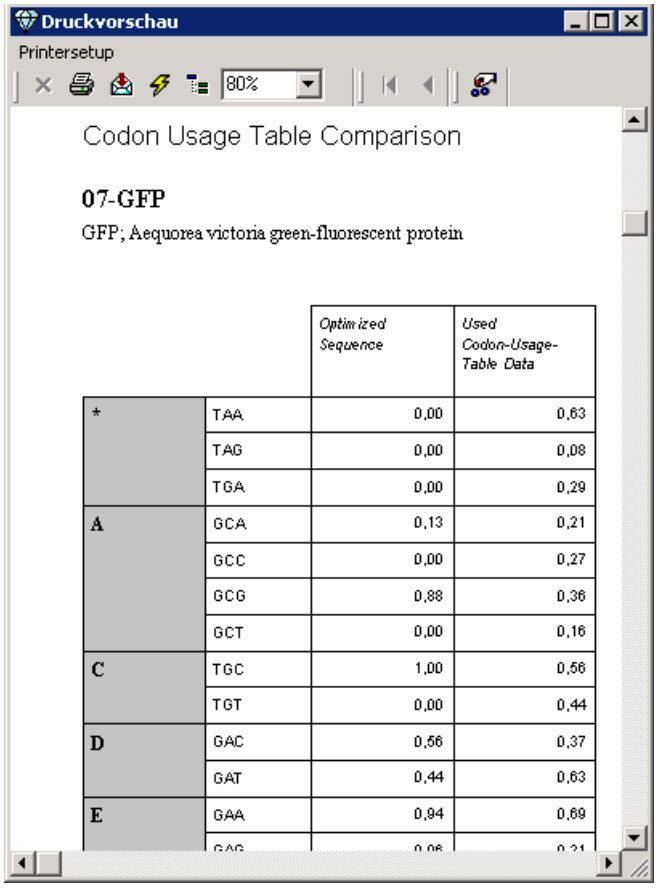


*Abb. C.3.6-2 Verlauf der Codon-Usage (als relative Adaptiveness w*100) entlang der Sequenz*



		Optimized Sequence	*Used Codon-Usage-Table Data*
*	TAA	0,00	0,63
	TAG	0,00	0,08
	TGA	0,00	0,29
A	GCA	0,13	0,21
	GCC	0,00	0,27
	GCG	0,88	0,36
	GCT	0,00	0,16
C	TGC	1,00	0,56
	TGT	0,00	0,44
D	GAC	0,56	0,37
	GAT	0,44	0,63
E	GAA	0,94	0,69

Abb. C.3.6-3 Ausschnitt aus einer Gegenüberstellung der Codon-Usage-Tabellen „optimierte Sequenz" / Expressionssystem

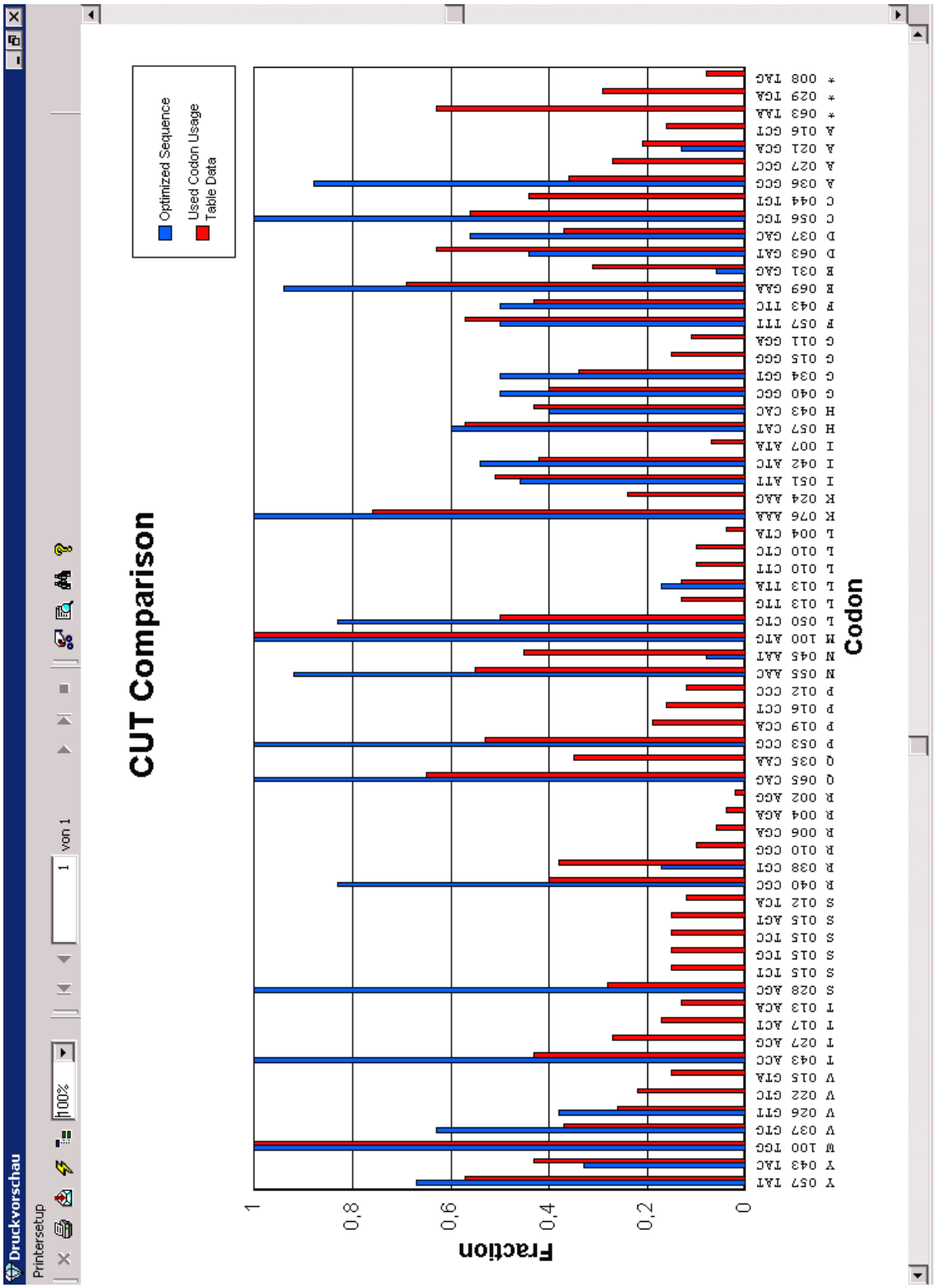


Abb. C.3.6-4 Graphische Gegenüberstellung der Kodonwahl der optimierten Sequenz und der für die Optimierung verwendeten Codon-Usage Tabelle.

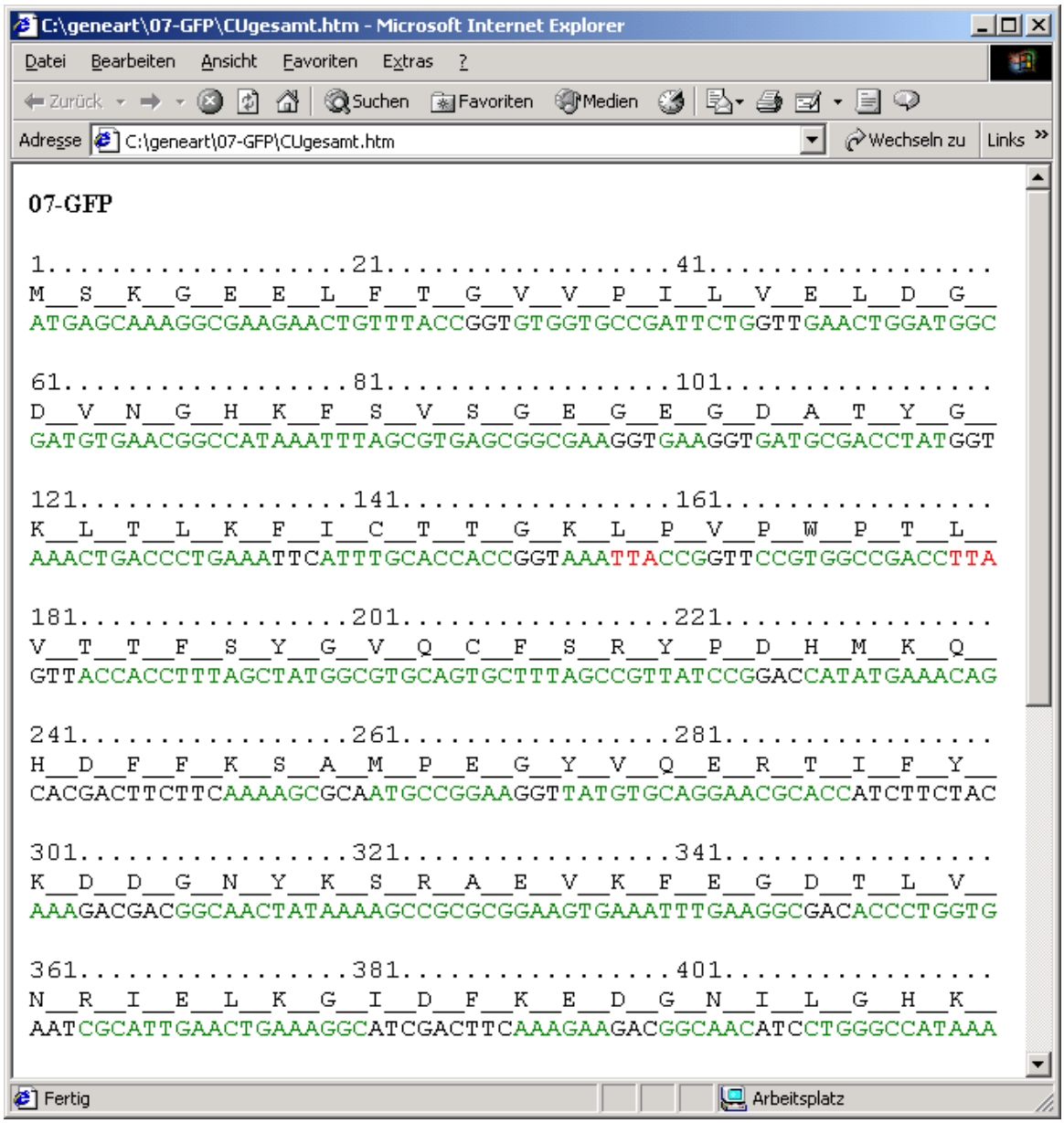


Abb. C.3.6-5 Darstellung der Sequenz unter Verwendung verschiedener Farben zur Symbolisierung der Kodonqualität: grün: relative Adaptiveness w>0.9, rot: w<0.5

Das Histogramm enthält jedoch keinerlei Informationen über die Verteilung der Kodonwahl-Güte entlang der Sequenz. Dazu erstellt die Funktion „Analyse->Codon Usage Verlauf" ein Liniendiagramm, auf dem die *w*-Werte der Kodons gegen die Basenposition aufgetragen sind. Will man die genaue Sequenzinformation mit einer qualitativen Darstellung der Kodonwahl verknüpfen, so kann dies mit der Funktion Analyse->Gesamtsequenz Kodonqualität (HTML) erreicht werden. Hierbei wird eine automatisch im WEB-Browser dargestellte HTML-Datei erzeugt, in der die Sequenz unter Verwendung unterschiedlicher Farben, welche die Kodonqualität symbolisieren, dargestellt wird. Schließlich kann eine Codon-Usage-Tabelle für das synthetische Gen berechnet werden und diese der Kodonwahl des Expressionssystems sowohl tabellarisch als auch grafisch gegenübergestellt werden. Die entsprechenden

Funktionen lassen sich über „Analyse->Cut-Vergleich Tabelle" und „Analyse->Cut-Vergleich graphisch" aufrufen.

C.3.7 GC-Verlaufsanalyse

Ein weiteres wichtiges Kriterium ist die Analyse des GC-Gehaltes. Dabei ist vor allem entscheidend, dass der GC-Gehalt an keiner Stelle einen extrem hohen oder niedrigen Wert annimmt. Dies lässt sich am einfachsten anhand des gleitenden Durchschnitts überprüfen. Dazu wird an der Abszisse der prozentuale GC-Gehalt eines um die aktuelle Basenposition zentrierten Fensters von beispielsweise 40 Nucleotiden Länge gegen die Basenposition aufgetragen. Die Fensterlänge lässt sich auf der Karteikarte „Graphen" durch den Anwender festlegen. Der Anwender kann die Erstellung des Diagramms über „Analyse->GC-Gehalt Verlauf" auslösen und erhält nach dessen Berechnung auch den durchschnittlichen GC-Gehalt der Gesamtsequenz im Meldungsfenster angezeigt.

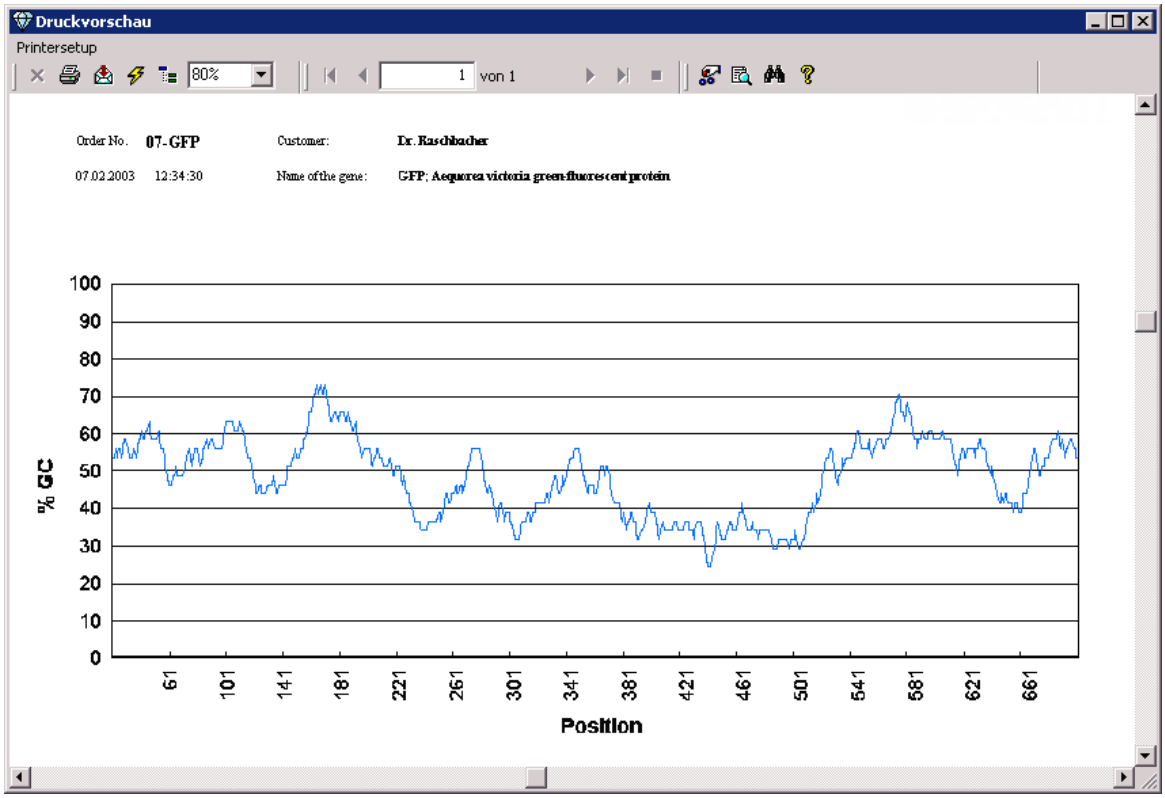


Abb. C.3.7-1 GC-Verlauf als gleitender Durchschnitt mit einem 40-Nucleotide-Fenster

C.3.8 DotPlot-Analyse

Während in der Problemstellen-Diagnose oder im Motivreport repetitive Elemente exakt als Alignment dargestellt werden, stellt ein Dotplot vor allem eine intuitive Möglichkeit dar, ähnliche Elemente innerhalb der Gesamtsequenz „auf einen Blick" zu erkennen.

Über das Menü „Analyse->Dotplot" öffnet sich dem Anwender ein weiteres Programmfenster, welches im unteren Bereich den Dotplot der DNA-Sequenz mit sich selbst enthält.

Die Größe des Dotplots kann über die Skalierungs-Dropdown-Liste variiert werden, so dass auch Dotplots längerer Sequenzen übersichtlich dargestellt werden können. Der Dotplot kann über die Kopieren-Schaltfläche in die Windows-Zwischenablage gestellt werden oder über „Speichern" als Bitmap-Datei abgelegt werden.

In der Regel lassen sich in einem nativen Dotplot kleinere zueinander ähnliche Sequenzbereiche nur schwer erkennen. Dies kann allerdings durch Anwenden eines geeigneten Filters erleichtert werden. Dazu wird über alle Diagonalen ein mehrere Nucleotide langes Fenster geschoben, in dessen Mitte ein Punkt gesetzt wird, falls innerhalb des Fensters eine definierte Mindestanzahl von Entsprechungen vorhanden ist. Auf diese Weise lassen sich sehr kleine repetitive Elemente oder Bereiche mit nur mäßig ausgeprägter Ähnlichkeit aus der Darstellung des Dotplots eleminieren, während einander stark ähnelnde Bereiche als durchgezogene und zur Hauptdiagonalen symmetrische diagonale Linien deutlich erkennbar werden. Um wieder den nativen Dotplot darzustellen, genügt ein Klick auf die „Dotplot"-Schaltfläche.

C.3.9 Blast-Analyse der DNA-Sequenz

Oftmals ist es notwendig zu überprüfen, ob die (Aminosäure-)Sequenz des synthetischen Gens zu natürlich vorkommenden Gensequenzen homologe Bereiche aufweist. So fordert z.B. die Food-and-Drug-Administration, dass bei humanen Gentherapieversuchen verwendete Vektoren keine zum humanen Genom homologe Bereiche aufweisen dürfen, um unerwünschte Rekombinationsereignisse sicher auszuschließen. Ebenso ist zu Vermeidung von RNA-Interference-Effekten eine Sequenzähnlichkeit zum Transkriptom des Expressionssytems zu vermeiden. Das WEB-Interface http://www.ncbi.nlm.nih.gov/BLAST/ der BLAST-Softwaresuite des National Center for Biotechnology Information (NCBI) bietet hierfür ideale Voraussetzungen. Um den Vergleich der Sequenz des synthetischen Gens mit den in der GenBank erfassten Sequenzdaten [Altschul 1990] möglichst einfach zu gestalten, wurde eine entsprechende HTTP-Schnittstelle in die GeneOptimizer-Software integriert [NCBI 2001].

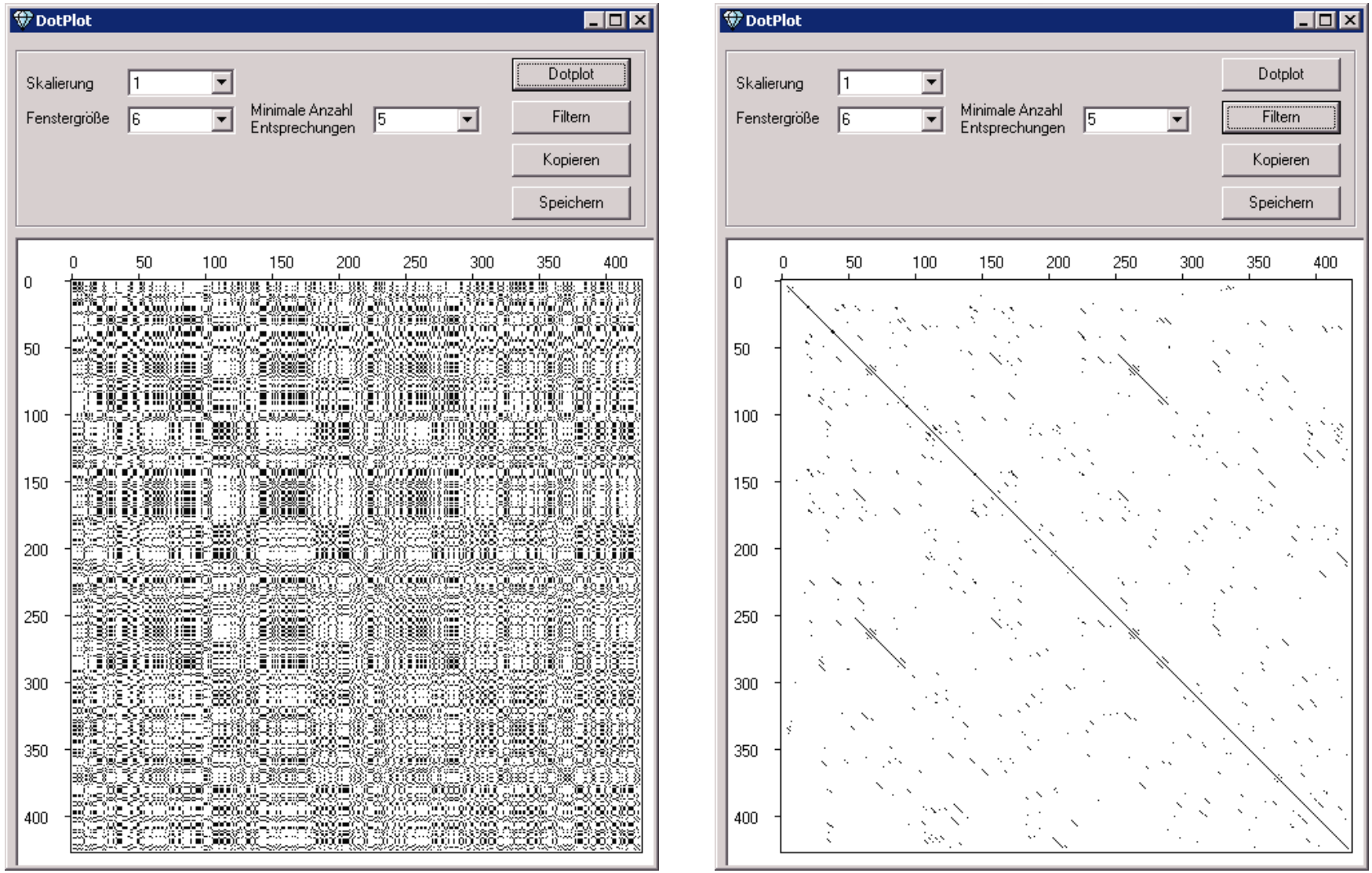


Abb. C.3.9-1 Dotplot einer Sequenz mit sich selbst. Die ca. 50 Nt lange Repetition ist erst nach Anwendung eines Filters (Fenstergröße 6 Nt., mindestens 5 Entsprechungen) deutlich zu erkennen.

Über die Menüfunktionen „Analyse-BLAST DNA“ und „Analyse->Blast Protein“ wird die DNA-Sequenz bzw. Aminosäuresequenz dem BLAST-Server, der entweder über das Internet erreicht oder aber auch lokal in ein Intranet eingebunden sein kann, übermittelt. Der Ablauf der Kommunikation zwischen GeneOptimizer-Rechner und BLAST-Server kann über das Meldungs-Fenster verfolgt werden. Nach Durchführung der BLAST-Suche kann der Anwender die Analyse über den automatisch geöffneten Standard-Webbrowser einsehen und die vom Server zur Verfügung gestellten weiteren Analysemöglichkeiten nutzen.

C.4 Optimierung

C.4.1 Wahl der Parameter und Durchführen der Optimierung

Die automatische Optimierung bildet das Kernstück der GeneOptimizer-Suite. Mit ihrer Hilfe kann der Anwender die seinen Vorgaben möglichst optimal entspechende DNA-Sequenz ermitteln.

Wie in Kap. B.2.3.1 detailliert beschrieben, wird zur Bewertung einer DNA-Sequenz eine komplexe Gewichtungsfunktion verwendet, die zahlreiche individuell gewichtbare Sequenzeigenschaften berücksichtigt. Die dazu notwendigen Parameter können im Eigenschaften-Fenster auf der „AutoOptim" –Dateikarte eingestellt werden.

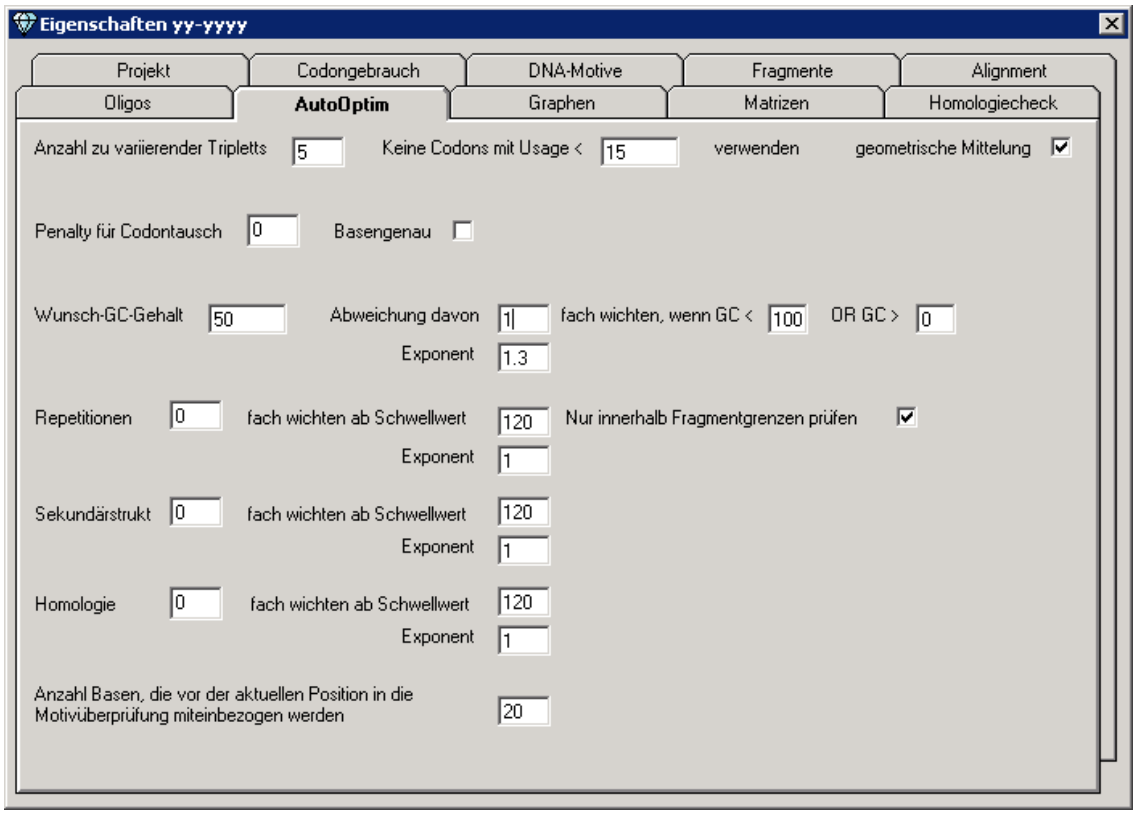


Abb. C.4.1-1 Einstellung der Parameter für die automatische Optimierung

Über die „Anzahl zu variierender Tripletts" bestimmt der Anwender, wie viele Kodons zu Bildung der Testsequenzen variiert werden. Hier gilt es, zeitliche und qualitative Anforderungen abzuwägen, da eine zu geringe Zahl zu einer schlechten Optimierung führt, die benötigte Rechenzeit mit einer Erhöhung des Wertes allerdings exponentiell ansteigt. Weiterhin werden für die Erstellung der Testsequenzen nur Kodons berücksichtigt, deren relative Angepasstheit über dem in dem Feld „keine Kodons mit Usage < verwenden" eingetragenen Wert liegt. Die durchschittliche Kodonwahl eines Sequenzabschnittes wird standardmäßig durch arithmetische Mittelung der W-

Werte der einzelnen Kodons ermittelt. Über das Hakenfeld „geometrische Mittelung" kann jedoch diese Form der Mittelwertbildung gewählt werden, die im Wesentlichen der Berechnung der relativen Angepasstheit nach Sharp und Li [Sharp 1987] entspricht.

In Einzelfällen kann es sinnvoll sein, die bestehende DNA-Sequenz möglichst unverändert zu belassen (d.h. die Kodonwahl nicht zu optimieren), wirklich kritische Sequenzmotive oder problematische Stellen aber dennoch zu eleminieren. Dazu können für den Austausch eines Kodons Strafpunkte vergeben werden, welche bei der Ermittlung des Gütescores abgezogen werden. Weiterhin kann differenziert werden, ob diese basenspezifisch oder kodonspezifisch vergeben werden, da ja abhängig von der Sequenz der beiden vertauschten Kodons ein bis drei Basen ausgetauscht werden. Bei der Berechnung der Gütefunktion fließt allerdings die in der gewählten Codon-Usage-Tabelle erfasste Güte der Kodons nach wie vor ein, so dass beim Austausch eines Kodons ein im Expressionssystem häufig genutztes bevorzugt eingesetzt wird.

Zur Anpassung des GC-Gehaltes muss zunächst ein Wunsch-GC-Gehalt definiert werden. Weiterhin können zwei Schwellwerte festgelegt werden, innerhalb derer der GC-Gehalt nicht in die Bewertungsfunktion einfließt. Sowohl die Bewertung des GC-Gehaltes als auch die der Repeats bzw. potentiellen Sekundärstrukturen (invers-komplementären Repetitionen) und der Homologie zu einer vorgegebenen Sequenz wird nach der allgemeinen Formel $g * Wert^{Exponent}$ vorgenommen, wobei *Wert* den „nativen" Score einer Sequenzeigenschaft, z.B. die prozentuale Abweichung vom Wunsch-GC-Gehalt oder den Alignmentscore darstellt. Der Gewichtungsfaktor g und der benutzte Exponent können vom Anwender in die entsprechenden Felder eingegeben werden. Wird der Gewichtungsfaktor gleich Null gesetzt, so fließt die entsprechende Eigenschaft nicht in die Gütefunktion ein und die zugehörigen Berechnungen werden auch nicht ausgeführt, was z.T. die Optimierung wesentlich beschleunigen kann. Sowohl bezüglich der Repetitionen als auch der Sekundärstrukturen und des Homologiekriteriums kann ein Schwellwert festgelegt werden (welcher sich auf den „nativen" Score bezieht), unterhalb dessen diese Sequenzeigenschaften nicht in die Gütefunktion eingehen. Die Sequenz, welche durch Berechnung des optimalen lokalen Alignmentscores auf zur Testsequenz homologe Bereiche überprüft wird, kann auf der „Homologiecheck"-Reiterkarte in ein Textfeld eingegeben werden. Wählt der Anwender die Option „Nur innerhalb Fragmentgrenzen prüfen", so nimmt die Software zunächst eine Grobunterteilung der Gesamtsequenz in überlappende Subfragmente vor. Bei der Optimierung werden dann nur Repetitionen vermieden, welche innerhalb eines Subfragmentes liegen würden. Dies ist sinnvoll, wenn (weiter auseinander liegende) Repetitionen im Expressionssystem

nicht stören, aber dennoch Probleme bei der Batch-Synthese vermieden werden sollen.

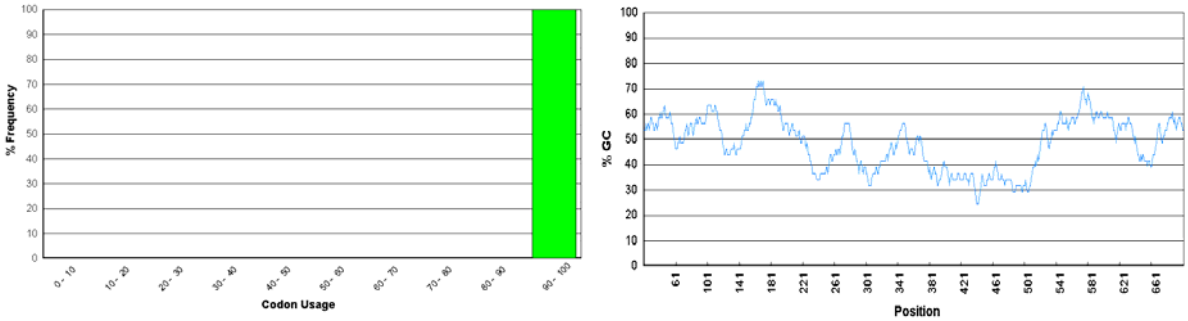
Sollen während der Optimierung bestimmte DNA-Motive vermieden oder in die Sequenz eingeführt werden, so müssen diese auf der „DNA-Motive"-Reiterkarte aus der Motivliste mit Hilfe der „Hinzufügen"-Schaltfläche in die Listen „auf diese Sites überprüfen" bzw. „Sites einführen" übernommen werden. Die Felder „Penalty" bzw. „Bonus" werden dabei zunächst automatisch mit dem unter „Gewichtung" in der Motivtabelle erfassten Wert gefüllt und stellen den Wert dar, der bei der Berechnung der Gütefunktion wie in Kap. B.2.3.5 beschrieben vom Score abgezogen bzw. hinzugezählt wird. Die Werte können jedoch in den beiden Listen auch individuell abgeändert werden. Durch die freie Wertvergabe ist es beispielsweise möglich, zwischen DNA-Motiven, welche unbedingt vermieden werden müssen, und DNA-Motiven, die zwar unerwünscht, aber in Abwägung gegen eine sehr schlechte Kodonwahl auch toleriert werden können, zu differenzieren.

Der Anwender kann den Fortgang der Optimierung am Bildschirm verfolgen, da die Einfügemarke an der Stelle positioniert wird, welche in der momentanen Iteration den Beginn der KDS darstellt. Auf einem dem Stand der Technik entsprechenden Computer kann eine Optimierung eines 1 kB-Genes mit einer KDS-Länge von fünf Kodons unter Berücksichtigung aller Sequenzeigenschaften in weniger als einer Minute abgeschlossen werden. Dadurch ist die Möglichkeit eines interaktiven Arbeitens mit der Software gewährleistet. Da bei ungünstiger Parameterkonstellation und längeren KDS jedoch auch Rechenzeiten im Minutenbereich auftreten können, kann die Optimierung durch eine auf dem Meldungsfenster befindliche „Abbrechen"-Schaltfläche jederzeit abgebrochen werden. Nach Fertigstellung der Optimierung wird dem Benutzer im Meldungsfenster die Anzahl evaluierter Testsequenzen mitgeteilt.

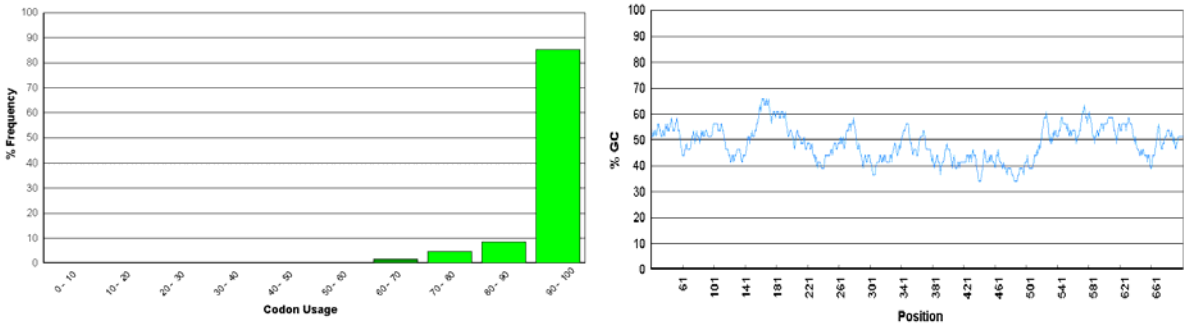
C.4.2 Beispiel einer zweiparametrigen Optimierung

Die über die Aminosäuresequenz-Importmöglichkeit erzeugte DNA-Sequenz des GFP ist zunächst rein in Hinblick auf die Kodonwahl optimiert. Der GC-Gehalt schwankt jedoch über den Verlauf der Sequenz sehr stark und erreicht an Extrempunkten über 70 bzw. unter 30 Prozent. Durch eine Variation in der Gewichtung der beiden Optimierungsparameter Kodonwahl und GC-Gehalt soll nun versucht werden, einen möglichst glatten GC-Gehaltsverlauf bei 50% unter gleichzeitiger Wahrung einer guten Kodonwahl zu erreichen. Die nachstehenden Beispiele stellen Optimierungen mit in der Parametergewichtung unterschiedlichen Bewertungsfunktionen dar.

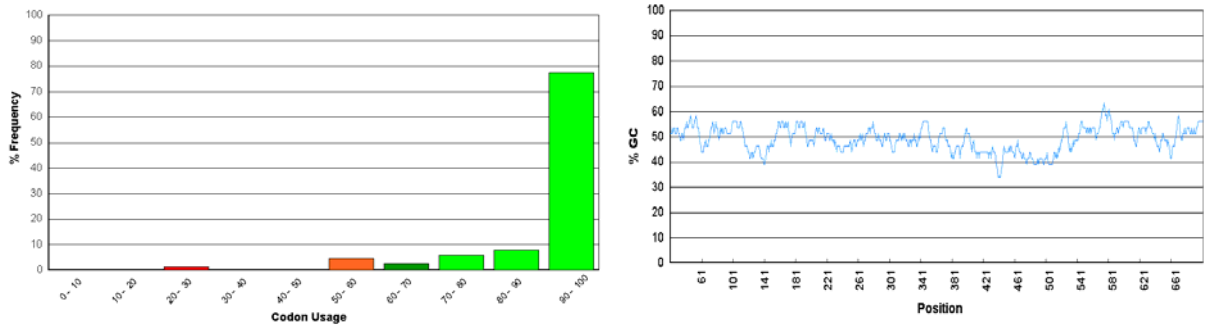
a; $Score = \langle CU \rangle$



b; $Score = \langle CU \rangle - \left| \langle GC \rangle - GC_{Wunsch} \right|^{1.3} \times 0.8$



c; $Score = \langle CU \rangle - \left| \langle GC \rangle - GC_{Wunsch} \right|^{1.3} \times 1.5$



d; $Score = \langle CU \rangle - \left| \langle GC \rangle - GC_{Wunsch} \right|^{1.3} \times 5$

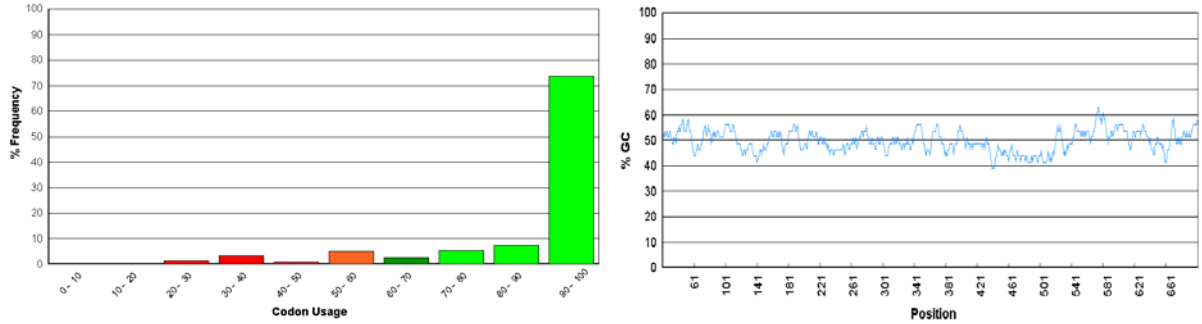


Abb. C.4.2-1 Analysen der DNA-Sequenz nach Optimierung mit verschiedenen Bewertungsfunktionen. Dabi bedeuten
*<CU>: arithmetischer Durchschnitt der w-Werte * 100 der KDS-Kodons*
<GC> : durchschnittlicher GC-Gehalt der letzten 35 Basen der Testsequenz in Prozent
GC_{Wunsch}: Angestrebter GC-Gehalt in Prozent
GC-Fenstergröße für Graphdarstellung: 40 Basen

In Abb. C.4.2-1a wird lediglich auf optimale Kodonwahl optimiert, was in einer sehr heterogenen und vom Ziel-Gehalt teilweise stark abweichenden GC-Verteilung resultiert. Die in Abb. C.4.2-1b gezeigte Optimierung verknüpft in idealer Weise eine Glättung des GC-Gehaltes um 50% mit einer guten bis sehr guten Kodonwahl. Abb. C.4.2-1c und C.4.2-1d verdeutlichen schließlich, dass eine weitere GC-Gehalts-Optimierung zwar möglich ist, aber mit einer stellenweise schlechten Kodonwahl erkauft werden muss.

C.5 Synthese

C.5.1 Unterteilung in Subfragmente

Steht die optimierte Sequenz des Synthetischen Genes fest, so unterstützt die GeneOptimizer-Suite den Anwender auch bei der Synthesevorbereitung. Nach Aufruf der Funktion „Synthese->Fragmentvorschläge" errechnet die Software die optimalen Schnittpositionen. Dabei beachtet die Software neben den in Kap. B.2.4.1 dargestellten Rahmenbedingungen auch die im Eigenschaften-Fenster auf der Reiterkarte „Subfragmente" vorgenommenen Einstellungen „Maximale Fragmentlänge" und die Liste auszuschließender Überhänge.

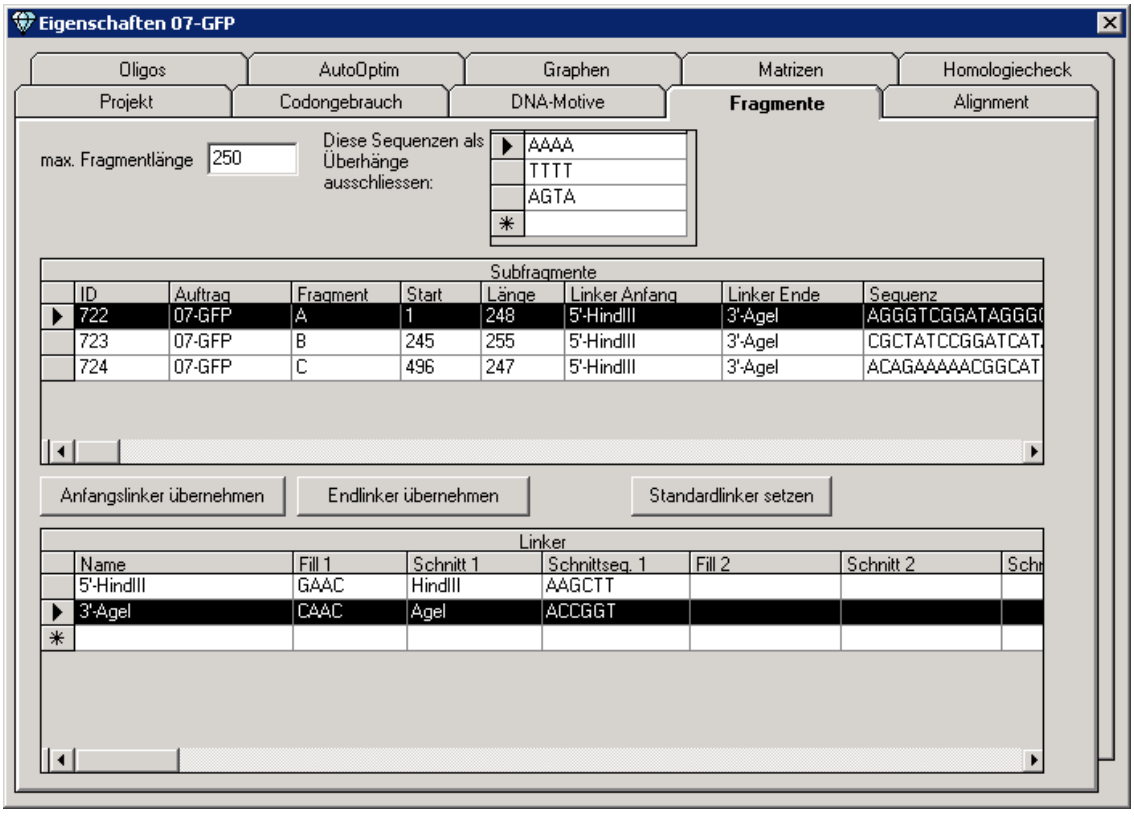


Abb. C.5.1-1 Die Subfragmentverwaltung

Die Schnittpositionen werden in der Sequenz als weiß markierte Base, welche auch das erste Nucleotid der Überlappsequenz repräsentiert, dargestellt. Über die „Nächste Fundstelle"-Schaltfäche kann der Anwender die vorgeschlagenen Schnittstellen anspringen und überprüfen und ggf. über die mit der rechten Maustaste zugänglichen Funktionen „aktuelle Markierung aufheben" und „Markierung Anfang" eine alternative Schnittposition bestimmen.

Über die Menüfunktion „Synthese->Fragmente generieren" werden die Fragmentsequenzen ermittelt und in die Datenbank geschrieben. Zugleich erscheinen sie in der Subfragmente-Tabelle auf der „Subfragmente"-Reiterkarte.

Die Fragmente werden grundsätzlich neben der Auftragsbezeichnung durch einen Buchstaben A-Z, der entsprechend der Reihenfolge in der Gesamtsequenz vergeben wird, identifiziert. Um einen Überblick über die Unterteilung der Gesamtsequenz zu erhalten, kann der Anwender über „Synthese->Gesamtsequenz Fragmente farbig" eine HTML-Darstellung der Gesamtsequenz veranlassen. In dieser werden die

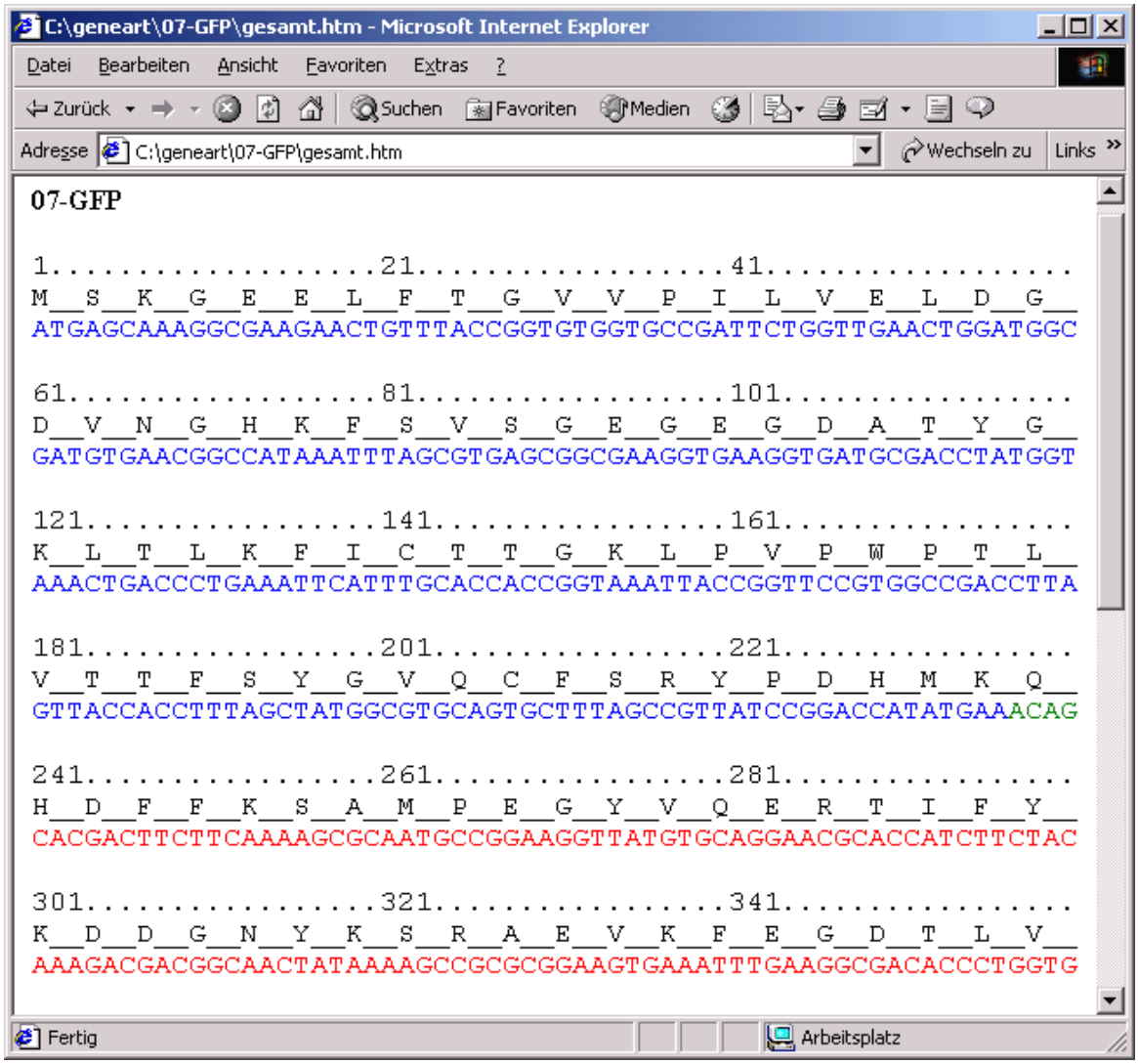


Abb. C.5.1-2 Ausschnitt aus der Gesamtsequenz mit farbiger Kennzeichnung der Subfragmente. Der Überlapp-Bereich ist grün gefärbt.

aufeinander folgenden Fragmentabschnitte abwechselnd blau und rot dargestellt, wobei die Überlappungen grün markiert sind. Die Subfragmente können nun noch mit geeigneten Linkersequenzen an beiden Enden versehen werden, um die notwendigen Klonierungsarbeiten zu ermöglichen. Ein Linker kann bis zu zwei Restriktionsschnittstellen so wie am Anfang und Ende zwischen den Schnittstellensequenzen „Dummy"-Füllsequenzen enthalten. Der Anwender kann beliebig viele Linkersequenzen in der entsprechenden Tabelle erfassen. Um eine Linkersequenz einem Subfragment zuzuweisen, markiert der Anwender ein oder mehrere Subfragmente in der Subfragmentliste, wählt einen Linker aus der Tabelle und weist diesen über die „Anfangslinker übernehmen" bzw. „Endlinker übernehmen"-Schaltflächen einem oder mehreren Subfragmenten zu.
In vielen Fällen kann die Gensynthese mit „Standardlinkern" durchgeführt werden. Dazu werden in die „Standard"-Spalte der Linkertabelle die Codes „e5","e3", „m3", „m5", „l3", „l5" eingetragen. e ,m und l stehen dabei für **e**rstes, **m**ittleres und **l**etztes Fragment, 3 und 5 bezeichnen das 3'- oder 5'-Sequenzende. Mit der „Standardlinker setzen"-Schaltfläche können nun mit einem Tastendruck allen Subfragmenten die korrekten Linker zugewiesen werden.

C.5.2 Aufspaltung in Oligonucleotide

Stehen die Sequenzen den Subfragmente einschließlich evtl. erforderliche Linkersequenzen fest, so können nun die Sequenzen der zur Synthese benötigten Oligonucleotide gemäß den unter „Materialien und Methoden" beschriebenen Algorithmen generiert werden.
Die dazu nötigen Parameter wie maximale Oligolänge, minimale Überlapp-Länge sowie angestrebte Schmelztemperatur der Hybridisierungsstrecken können auf der Karteikarte „Oligos" erfasst werden. Über „Synthese->Oligos generieren" löst der Anwender den Zerlegungsprozess aus, dessen Fortschritt er über das Meldungsfenster verfolgen kann. Neben den für die Ligation benötigten Oligonucleotiden generiert die Software auch die Sequenzen für die PCR-Primer, welche zur Amplifizierung der durch die Ligation gebildeten Fragmente benötigt werden.
Nicht immer kann jedoch die angestebte Schmelztemperatur eingehalten werden bzw. die Software erkennt ein mögliches Syntheseproblem durch Fehlhybridisierungen von Oligonucleotiden. In diesem Fall wird im Meldungsfenster eine Warnung bei zu niedriger Schmelztemperatur oder im Falle einer möglichen Fehlhybridisierung deren Alignment unter Angabe der betroffenen Oligonucleotide ausgegeben.

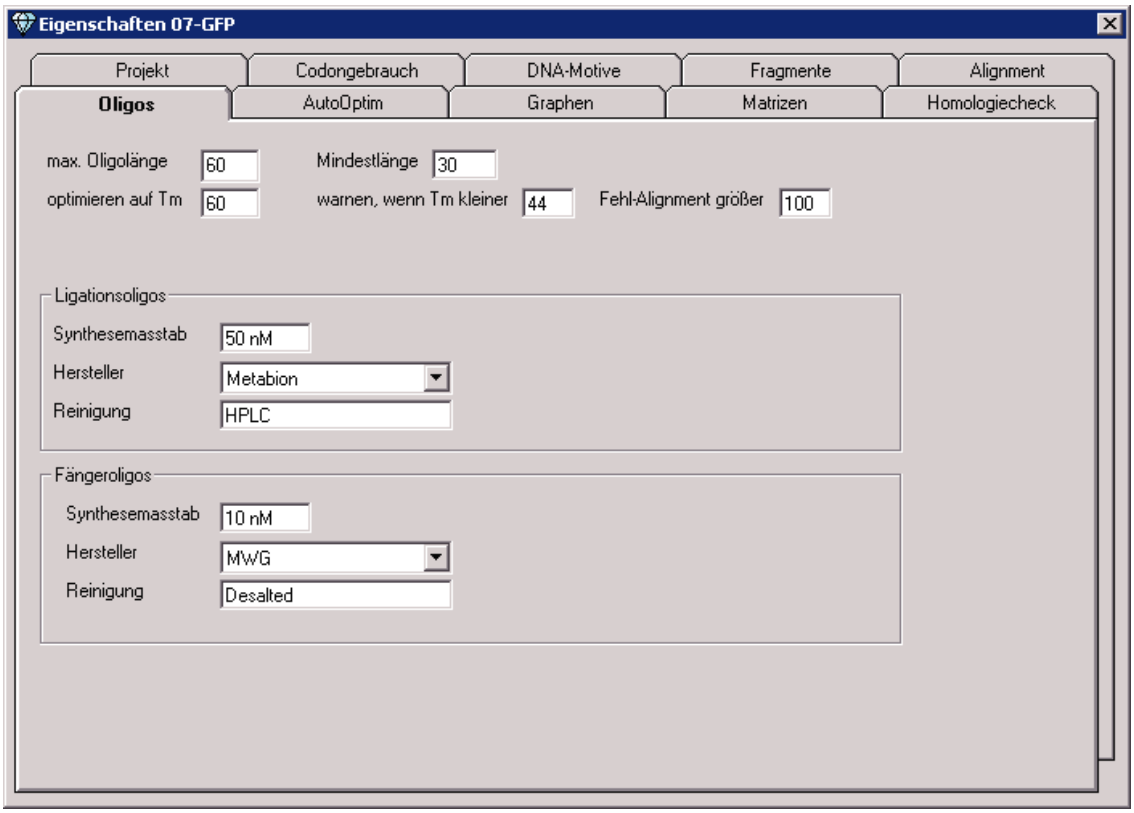


Abb. C.5.2-1 Einstellung der Parameter für die Oligozerlegung

Die entsprechenden Warnschwellen können ebenfalls auf der „Oligo"-Dateikarte eingegeben werden (Welche Schmelztemperatur muss unterschritten werden bzw. welchen Score muss das Alignment einer Fehlhybridisierung haben, damit eine Warnmeldung generiert wird.).

Die Funktion „Synthese->Subfragmente Oligos farbig" generiert eine HTML-Ausgabe der Subfragment-Sequenzen als dsDNA, in welcher die Ligationsoligonucleotide abwechselnd blau und rot und die Fängeroligonucleotide orange und grün dargestellt sind. Freie Stellen in der Antisense-Sequenz, welche nicht durch Fängeroligos abgedeckt sind, sind schwarz dargestellt. Zwei Linen unterhalb der Sequenz lassen durch farbige Markierung erkennen, welcher Bereich der Sequenz Linkersequenzen darstellt und wie lange die zum Fragment gehörenden PCR-Primer sind.

Schließlich können die generierten Oligosquenzen über „Synthese->Oligobestellung" tabellarisch in 5'->3' - Orientierung als Bericht ausgegeben werden. Dabei wird hinsichtlich Synthesemaßstab, Aufreinigung, Phosphorylierung und Oligohersteller zwischen Ligations und Fängeroligonucleotiden unterschieden. Die verwendeten Parameter können ebenfalls auf der Oligokarteikarte festgelegt werden. Über die Exportfunktion der Seitenvorschau kann die Liste ggf. z.B. als Excel-Datei abgespeichert werden, die alle vom Oligohersteller benötigten Informationen enthält und diesem direkt zugesendet werden kann.

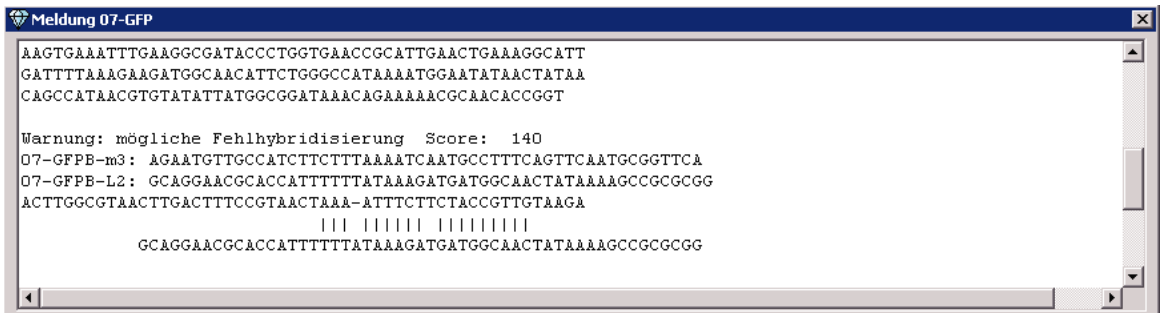


Abb. C.5.2-2 Ausgabe von Warnungen bezüglich möglicher Fehlhybridisierungen oder zu niedriger Schmelztemperatur im Meldungsfenster.

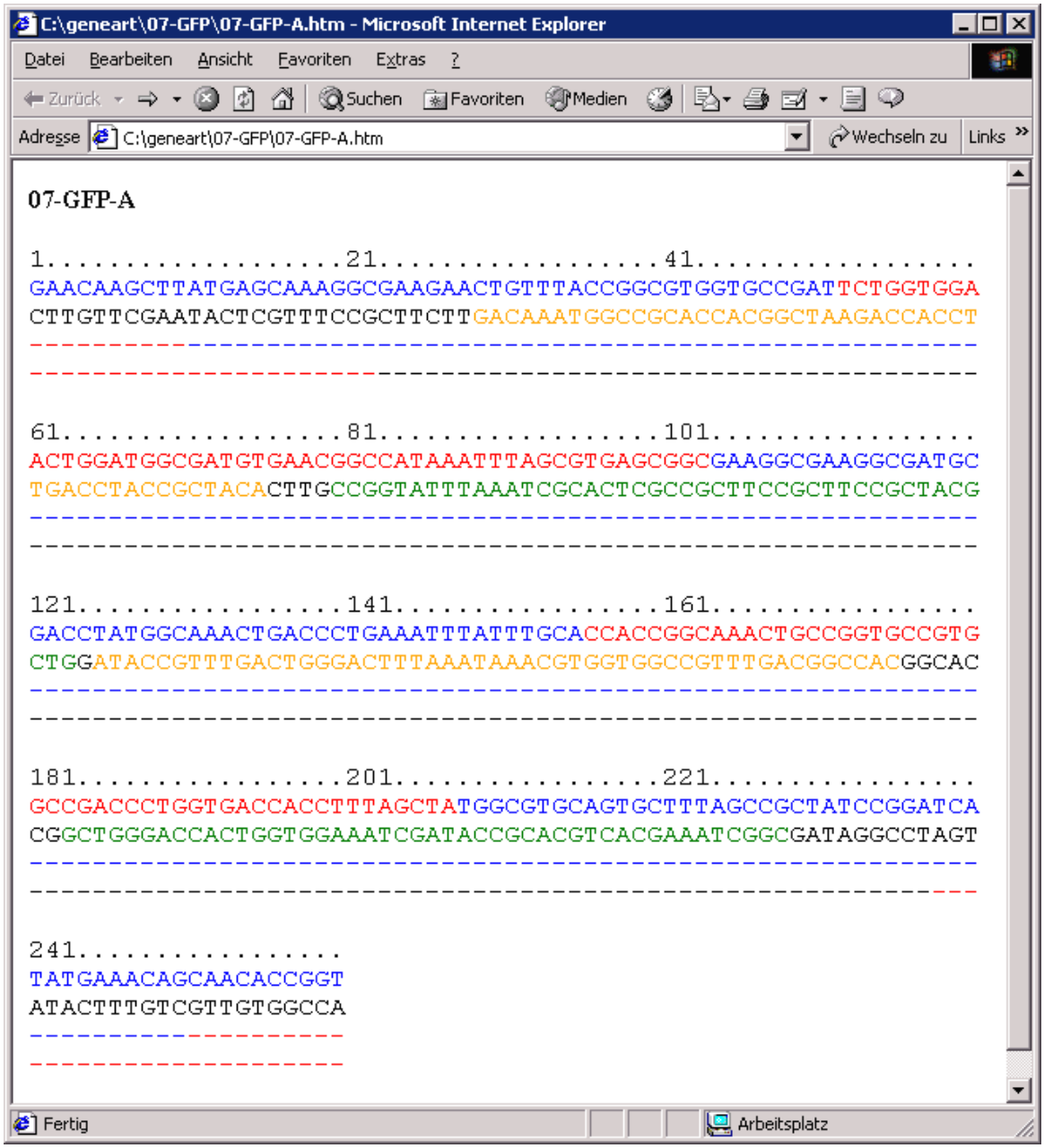


Abb. C.5.2-3 Subfragment mit farbiger Oligounterteilung: blau/rot - Ligationsoligos, orange/grün –Fängeroligos. Die unterhalb der Sequenz befindlichen Linien symbolisieren die Länge der Linker bzw. der PCR-Primer

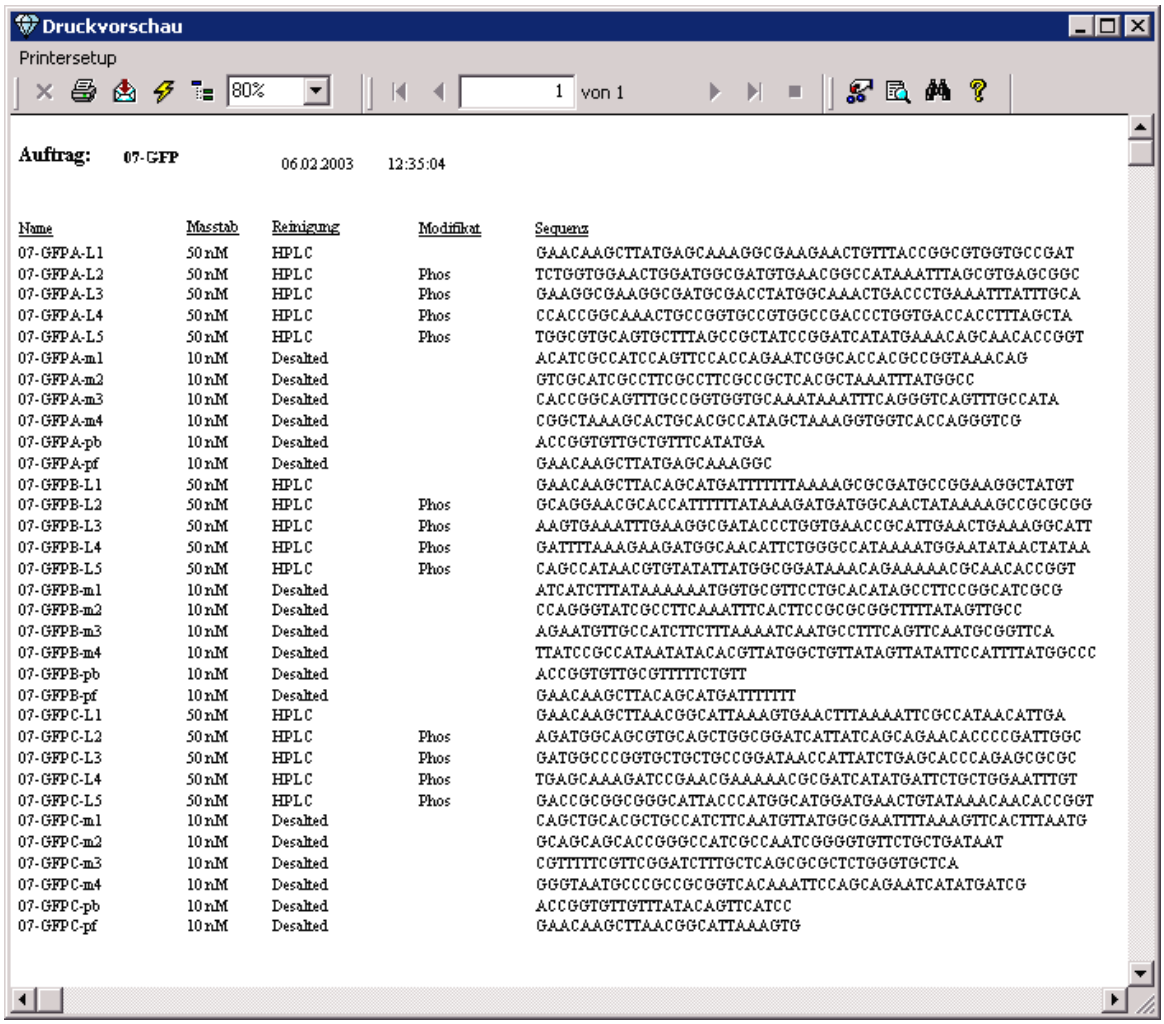


Abb. C.5.2-4 Oligobestellung

C.5.3 Finaler Sequenzvergleich

In der Regel wird vom Finalklon eines synthetischen Gens gefordert, dass seine Sequenz 100prozentig der Vorgabesequenz entspricht. Allenfalls stille Mutationen können in Einzelfällen toleriert werden. Um die Sequenzidentität zu überprüfen und ggf. in geeigneter Weise zu dokumentieren, kann der Anwender sich der Funktion „Extras->Sequenzvergleich" bedienen.

Das „Sequenz 1: Vorgabesequenz in Datenbank"-Textfeld des zugehörigen Dialogs ist standardmäßig bereits mit der aktuellen DNA-Sequenz des Sequenzeditors belegt. Um die Sequenzvergleichsroutine möglichst flexibel einsetzen zu können, kann der Anwender hier jedoch auch eine andere Sequenz seiner Wahl einfügen. Das „Sequenz 2" Textfeld ist für die Sequenz des Finalklons vorgesehen. Diese wird im Regelfall die Vorgabesequenz sowie einen kürzeren Sequenzbereich davor und danach umfassen, welcher mitsequenzierte Vektorsequenz darstellt. Daher wird über die „Align"-Schaltfläche ein „End-Space-Free"-Global-Alignment Algorithmus aufgerufen, der diese Tatsache bei der Durchführung des Alignments berücksichtigt. Wird das „Für

Vorgabesequenz IUPAC-Nomenklatur beachten" gehakt, werden in der Vorgabesequenz enthaltene degenerierte und mit IUPAC-Symbolen verschlüsselte Basenpositionen korrekt interpretiert. Letztere finden vor allem Verwendung, wenn die Vorgabesequenz eine Genbank darstellt, die an bestimmten Positionen randomisiert ist. Somit kann auf einfache Weise überprüft werden, ob der Finalklon ein Element der Genbank darstellt.

Nach Berechnung des Alignments enthält die „Abweichungen"-Auflistung alle festgestellten und als Insertionen, Deletionen sowie Substitutionen klassifizierten Unterschiede zwischen den beiden Sequenzen. Indem der Anwender eins der Auflistungselemente auswählt, kann er den zugehörigen Alignmentausschnitt im „Alignment"-Feld darstellen. Zugleich wird im Sequenzeditor die Einfügemarke an die entsprechende Basenposition gesetzt. Dies ermöglicht es dem Anwender, unmittelbar zu erkennen, ob es sich bei einer Substitution um eine stille Mutation handelt (d.h. die codierte Aminosäuresequenz bleibt unverändert) und ob in diesem Fall eventuell ein sehr „schlechtes" Kodon in Bezug auf die Kodonwahl Verwendung findet.

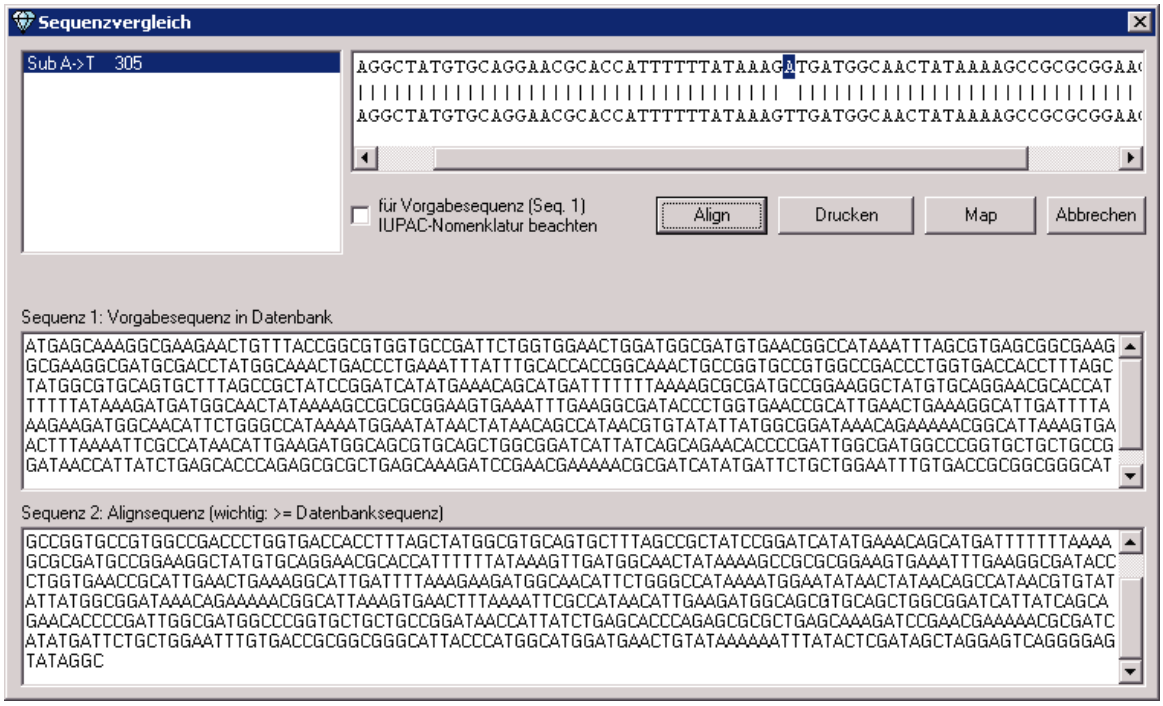


Abb. C.5.3-1 Sequenz-Alignment

Über die „Drucken"-Schaltfläche hat der Anwender die Möglichkeit, sich das komplette Alignment der beiden Sequenzen im Druckvorschau-Fenster anzeigen zu lassen und dieses dann beispielsweise auch auszudrucken. Über die „Map" Funktion lässt sich die Alignsequenz mit zugehöriger Motivannotation ausdrucken. Dies ist für den Anwender besonders hilfreich, da hierbei auch die für die Klonierung verwendeten Restriktionsschnittstellen annotiert werden können.

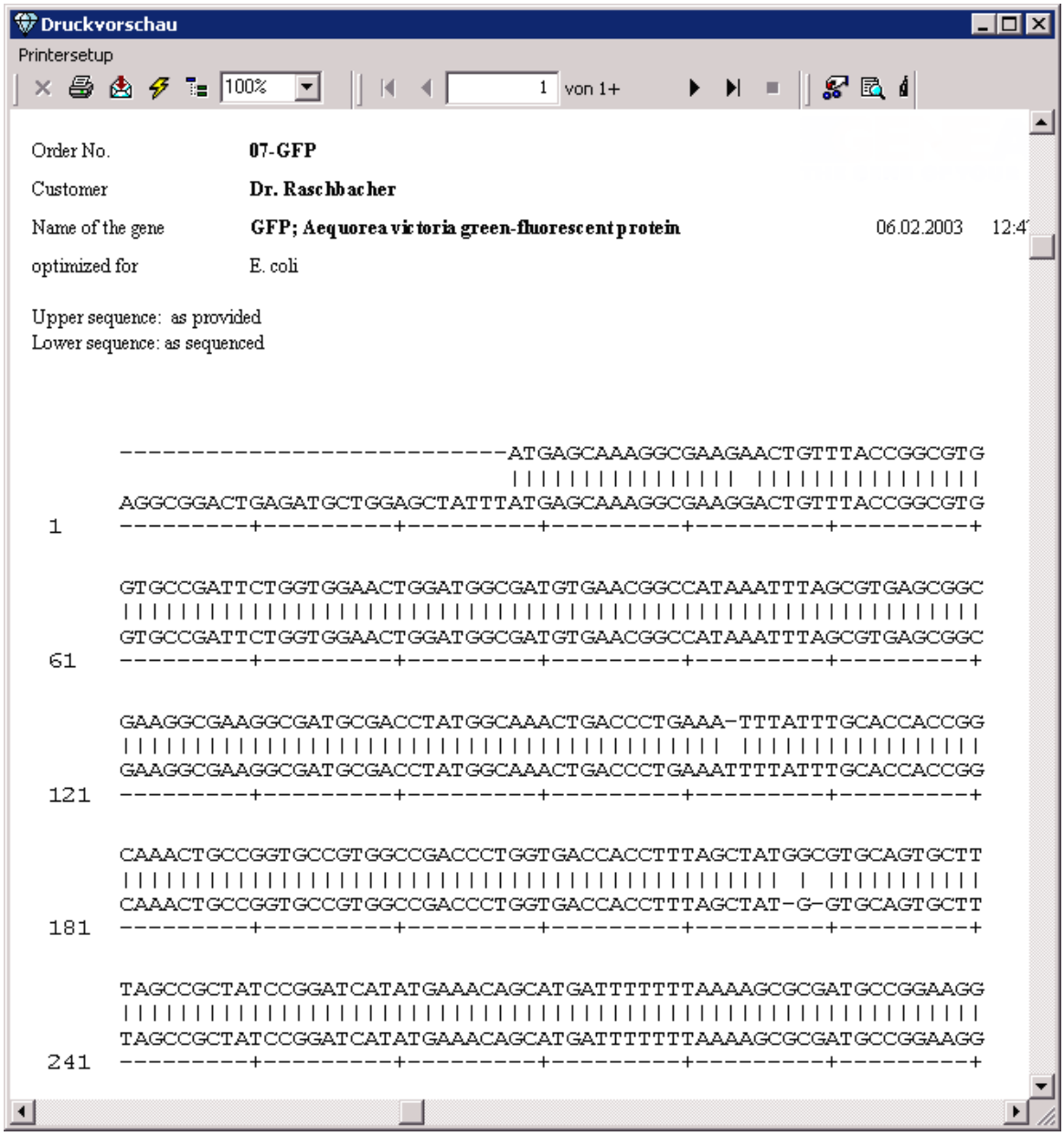


Abb. C.5.3-2 Ausdruck des Alignments der „Vorgabesequenz" (in der Datenbank befindliche Sequenz des synthetischen Gens) und der „Alignsequenz" (sequenzierter Vektor & Insert)

C.6 Analyse der Klonsequenzen

Abhängig von der Oligoqualität und der Länge der Subfragmente müssen unter Umständen mehrere Dutzend Klonsequenzen eines Subfragmentes evaluiert werden, bis ein Klon mit 100%ig korrekter Sequenz gefunden wird. Diese Arbeit kann durch das Programmmodul „Sequenze Analyzer" wesentlich erleichtert werden, da es die vom Sequenzer gelieferten Klonsequenzen automatisch analysiert und bewertet, so dass der Anwender in vielen Fällen nur noch die am besten bewertete Sequenz einer finalen visuellen Endkontrolle unterziehen muss.

C.6.1 Durchführen der Analysen

Die Sequenzdateien („.ab1"-Dateien) werden üblicherweise in einem nach der Auftragsbezeichnung benannten Unterverzeichnis eines allgemeinen „Sequenzierungen"-Verzeichnis abgelegt. Der Dateiname enthält dabei sowohl die Auftrags- und Subfragmentbezeichnung in der Form „003A", als auch die Bezeichnung des Klons.
Um die vorhandene Sequenzdateien der Klone eines bestimmten Subfragments zu analysieren, wählt der Anwender aus der Auftragsliste das entsprechende Projekt aus. Wird in der nun mit den zu dem gewählten Auftrag gehörenden Subfragmentkennungen gefüllten Fragmentliste das gewünschte Fragment ausgewählt, erstellt das Programm die passende Dateisuchmaske. Dieses Textfeld ist jedoch frei editierbar, so dass ggf. eine abweichende Nomenklatur der Dateien berücksichtigt werden kann.
Das durchsuchte Verzeichnis kann im Textfeld „Suchpfad" angegeben werden. Durch Betätigen der „Wählen"-Schaltfläche gelangt der Anwender zu einem Dialog, der eine vereinfachte Auswahl des gewünschten Verzeichnisses gestattet: Mit der oberen DropDown-Liste wird das gewünschte Laufwerk gewählt, anschließend kann in der unteren hierarchischen Darstellung ein Verzeichnis markiert werden.

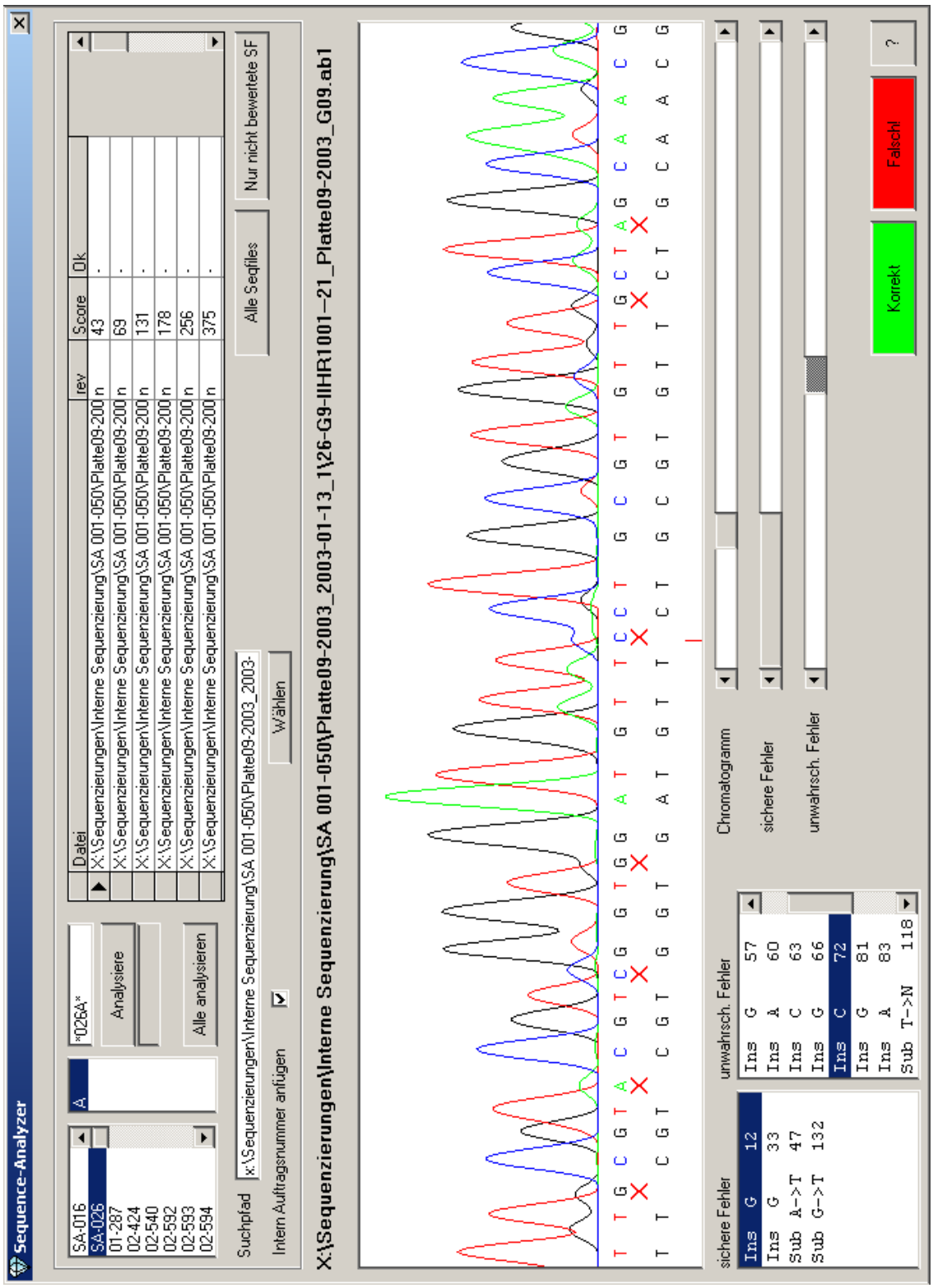


Abb. C.6.1-1 Arbeitsbereich des SequenceAnalyzers.

Ist das Feld „Intern Auftragsnummer anfügen" gehakt, so fügt die Software intern an den Suchpfad die Auftragsbezeichnung als Unterverzeichnis an, so dass der endgültige Dateisuchausdruck beispielsweise „C:\Sequenzierungen\05-003*003A*" lautet.

Nach Betätigen der „Analysiere"-Schaltfläche analysiert die Software alle dem Suchausdruck entsprechenden Dateien, welche noch nicht in einem früheren Analysedurchlauf bewertet wurden. Jedoch kann durch Wählen der Option „Neuevaluierung bereits analysierter Dateien erzwingen" eine Neubewertung auch schon in der Datenbank erfasster Dateien erreicht werden. Dies ist vor allem sinnvoll, wenn die Bewertungsparameter auf dem „Einstellungen"-Dialog geändert wurden und der Score aller Sequenzdateien mit den geänderten Parametern neu berechnet werden soll.

Ist die Analyse abgeschlossen, werden die untersuchten Dateien in der Sequenzdatei-Tabelle nach ihrer Güte, d.h. Sequenzkorrektheit, geordnet aufgelistet. Die Bewertung erfolgt wie in Kap. B.2.5 dargestellt, so dass ein möglichst niedriger Wert in der Spalte „Fehler" eine Sequenz mit wenigen (möglichen) Fehlern charakterisiert. Der Score wird dabei unter Verwendung der Gewichtungsparameter errechnet, welche auf dem „Einstellungen"-Fenster eingegenben wurden. In der „rev"-Spalte wird ein „n" für nein oder ein „j" für ja eingetragen, je nachdem ob die Sequenz mit einem forward-Primer oder einem reverse-Primer sequenziert wurde, d.h. die Sequenziersequenz mit der reverse-komplementären Sequenz der in der Datenbank gespeicherten Subfragment-Zielsequenz aligned wurde.

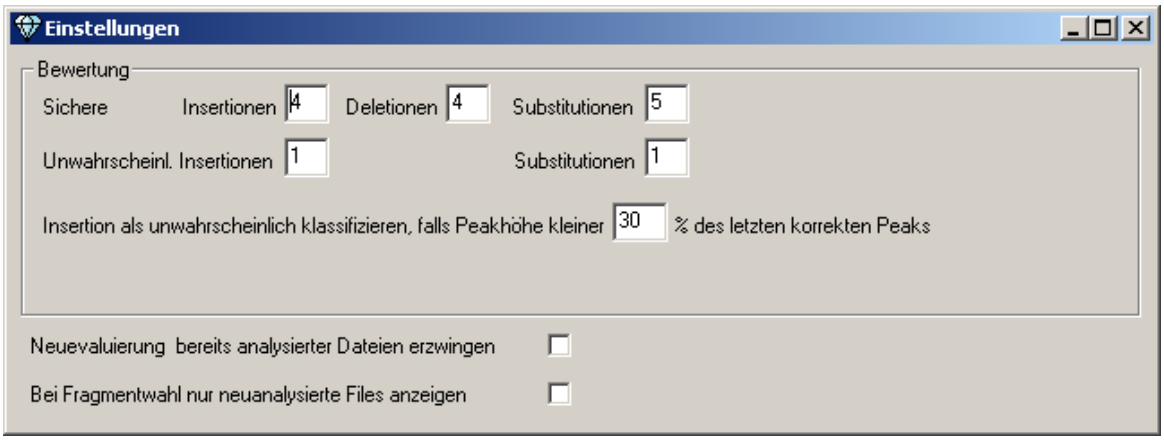


Abb. C.6.1-2 „Einstellungen"-Dialog

Wenngleich die Analyse einer Sequenzdatei auf einem dem Stand der Technik entsprechenden Rechner in weniger als zwei Sekunden abgeschlossen ist, kann es bei einer Vielzahl von zu analysierenden Fragmenten und Klonsequenzen zu unangenehmen Wartezeiten während der Arbeit mit dem Programm kommen. Daher bietet die Funktion „Alle analysieren" die Möglichkeit, automatisch alle vorhandenen

Sequenzdateien der in der Auftragsliste aufgeführten Projekte zu analysieren. Auf diese Weise können nach Abschluss der Analysen die Ergebnisse ohne weitere Verzögerungen begutachtet werden.

C.6.2 Ansicht der Analyse eines bestimmten Sequenzfiles

Wählt der Anwender einen Auftrag und ein dazugehöriges Subfragment aus den Listenfeldern aus, so werden in der Dateien-Tabelle alle im letzten Analysevorgang untersuchten Sequenzdateien aufgelistet. Jedoch können mit der „Alle Seqfiles"-Schaltfläche auch sämtliche in der Datenbank gespeicherten (und bei früheren Analysen bewerteten) Dateien aufgelistet werden.
Wählt der Anwender eine Datei aus der Tabelle aus, so wird das Sequenzelektropherogramm mit den vom Basecaller benannten Basen angezeigt. Unterhalb dieser Sequenz befindet sich das Alignment der in der Datenbank gespeicherten Zielsequenz des Subfragmentes. Mit Hilfe des „Chromatogramm"-Schiebereglers kann der Anwender den gezeigten Ausschnitt kontinuierlich verschieben. Der Name der gezeigten Sequenzdatei wird oberhalb des Elektropherogramms dargestellt.
Wie in Kap. B.2.5 dargestellt, unterscheidet der SequenceAnalyzer zwischen „sicheren" und „unwahrscheinlichen" Fehlern. Diese werden unter Angabe der Art des Fehlers (Del, Ins, Sub mit Angabe der relevanten Basen) und der Basenposition in zwei Listenfeldern aufgeführt.
Markiert der Anwender einen Eintrag, so wird automatisch das Elektropherogramm so verschoben, dass die entsprechende Problemstelle in der Mitte des gezeigten Ausschnitts dargestellt ist. Um die Art des Fehlers optisch leicht erkennbar darzustellen, wird zwischen den beiden Basenseqenzen in einer etwas größeren Schriftart dargestellt, welche Veränderungen notwendig wären, um die fehlerhafte Klonsequenz zu berichtigen. Im Falle von Substitutionen ist dies die korrekte Base, bei Insertionen symbolisiert ein großes „X", dass die entsprechende Base aus der Klonsequenz gestrichen werden müsste. Deletionen werden durch die fehlende Base angezeigt, wobei eine Verbindungslinie den genauen Ort anzeigt, an dem die Base insertiert werden müsste. Auf diese Weise kann der Anwender visuell anhand des Elektropherogramms leicht erkennen, ob ein angezeigter Fehler tatsächlich vorliegt oder eine Fehlinterpretation durch den Basecaller vorliegt.

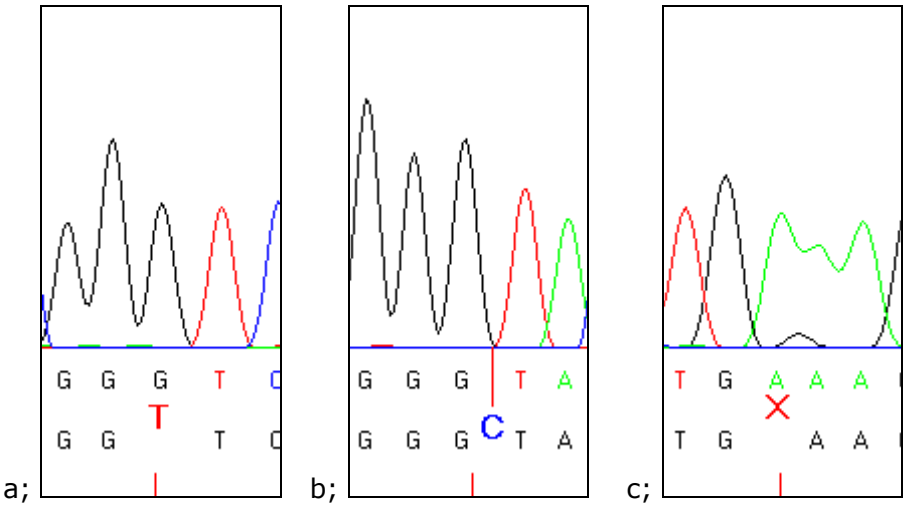


Abb. 6.2-1 Sichere Fehler: a;Substitution „T->G", b;Deletion „C", c;Insertion „A"

Neben der Auswahl eines Fehlers in den beiden Listenfeldern hat der Anwender auch die Möglichkeit, mit den beiden Schiebereglern „sichere Fehler" und „unwahrscheinliche Fehler" bequem von einer Problemstelle zur nächsten zu springen.

In der Praxis wird der Anwender zunächst die Sequenzdatei visuell überprüfen, welche den niedrigsten Fehlerscore aufweist, und dann zuerst die eventuell angezeigten „sicheren Fehler" überprüfen.

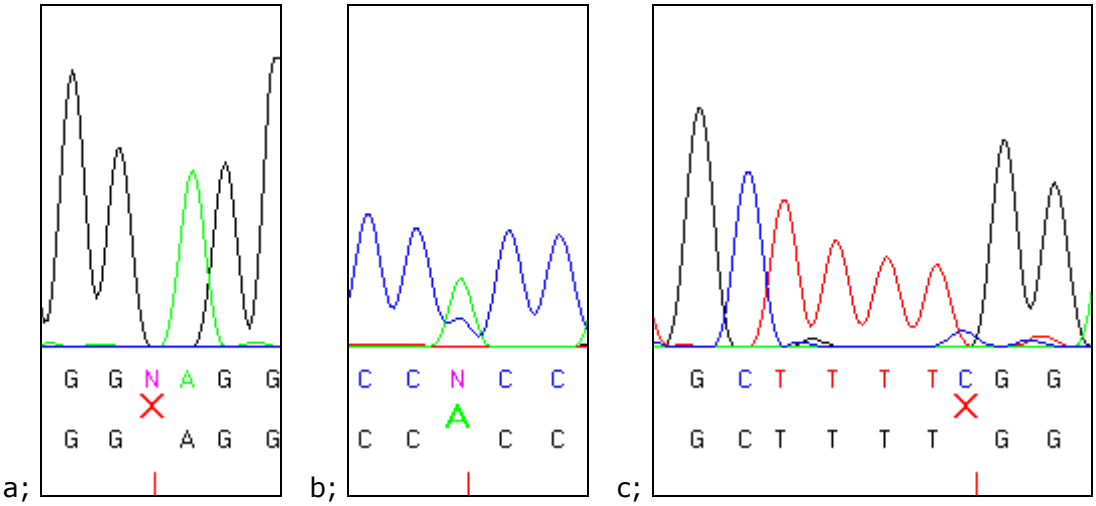


Abb. 6.2-2 unwahrscheinliche Fehler: a; Insertion „N", b; nicht sicher erkannte Base A, c;Insertion eines „C"

Liegen keine solchen vor oder liegt hierbei eine softwareseitige Fehlinterpretation vor, ist es dennoch empfehlenswert im Rahmen einer visuellen Endkontrolle auch die als „unwahrscheinlich" klassifizierten Fehler zu inspizieren.

Ist schließlich eine fehlerfreie Sequenz gefunden, kann der zugehörige Klon anhand des Dateinamens indentifiziert werden.

Mit den Schaltflächen „Richtig“, „Falsch“ und „?“ kann der Anwender eine analysierte Sequenzdatei endgültig bewerten. Das Fragezeichen steht dabei für den Fall, dass eine Datei zwar als gesehen gekennzeichnet werden soll, aber kein endgültiges Urteil über die Sequenzkorrektheit getroffen wurde. Nach Betätigen der entsprechenden Schaltflächen wird die Beurteilungin die „Ok“-Spalte übernommen und automatisch die nächste Sequenzdatei aufgerufen.

D Diskussion und Ausblick

D.1 Diskussion

Mit der GeneOptimizer-Suite und dem SequenceAnalyzer liegt ein benutzerfreundliches Softwarepaket vor, welches derzeit alle wesentlichen Schritte der Gensynthese, angefangen vom Design über die Produktionsunterstützung bis hin zur Sequenzauswertung unterstützt. Dabei konnte im Rahmen dieser Arbeit ein neuartiger Optimierungsalgorithmus entwickelt werden, der wesentliche Vorteile gegenüber den literaturbekannten Verfahren bietet. Viele Charakteristika einer DNA-Sequenz lassen sich als vollständig (z.B. DNA-Motive wie RE-Sites) oder zumindest teilweise lokale (lokaler GC-Gehalt, Haarnadelschleifen, Homologiebereiche zu anderen Sequenzen, etc.) Phänomene beschreiben, welche sich in einem Sequenzfenster von wenigen Kodons abspielen. Mit Hilfe des „gleitenden Kombinationsfensters" kann erstmals vollautomatisch ein synthetisches Gen entworfen werden, dessen Sequenz den vom Anwender vorgegebenen Zielen in Bezug auf die genannten Eigenschaften in nahezu optimaler Weise entspricht. Im Gegensatz dazu werden sich rein stochastische Verfahren der (lokal) optimalen Sequenz in den meisten Fällen nur annähern können. Dies ist besonders offensichtlich bei der gewünschten Einführung bestimmter Sequenzmotive, da hierbei die Veränderung eines einzigen Kodons über das Vorkommen des Motives in der Sequenz entscheiden kann. Andererseits kann bei der durch unser Verfahren idealerweise betriebenen vollständigen Durchtestung des KDS-Kombinationsraumes auch die Sequenz gefunden werden, die ein durch eine Häufigkeitsmatrix definiertes DNA-Motiv durch Eleminierung der für die Aktivität wichtigsten Basen am effektivsten ausschaltet (bzw. eine ideale Kompromisslösung bei Einbeziehung anderer Kriterien gefunden werden).

Dehnt man die Betrachtung auf Kriterien aus, welche sich nicht als rein lokale Phänomene beschreiben lassen, wie beispielsweise Repetitionen, scheint die Anwendung stochastischer Methoden, die in jedem Optimierungsschritt bereits die Gesamtsequenz variieren und bewerten können, Vorteile aufzuweisen. Würde die Rückübersetzung einer Aminosäuresequenz bei Verwendung optimaler Kodons zu einer DNA-Sequenz führen, welche zwei am Anfang bzw. im Endbereich liegende identische Bereiche aufweist, so ist ein stochastischer Algorithmus in der Lage, diese Repetition durch die Einführung synonymer „schlechterer" Kodons in beiden der ursprünglich zueinander homologen Sequenzabschnitte zu eleminieren. Das in 5'->3' gleitende Kombinationsfenster würde dagegen die optimale Kodonwahl im ersten

Abschnitt belassen und die Entstehung einer Repetition durch Verwendung schlechterer Kodons im endständigen Bereich vermeiden.
Diese auf den ersten Blick ungünstige Häufung schlechterer Kodons im zweiten Bereich (und die ausschließliche Verwendung optimaler im ersten Abschnitt) ist jedoch tatsächlich im biologischen System vermutlich die bessere Lösung. Experimentell konnte gezeigt werden, dass für die optimale Expression von Proteinen die Verwendung optimaler Kodons besonders am 5'-Ende der RNA entscheidend ist, während weiter stomabwärts gelegene schlechtere Kodons eher toleriert werden [Rosenberg 1993, Goldman 1995].

Nach der Ermittlung der optimierten Sequenz erlauben vielfältige integrierte Analysewerkzeuge dem Anwender, in Hinblick auf die Optimierung einen Vorher-Nachher-Vergleich durchzuführen und somit interaktiv die Optimierungsparameter so anzupassen, dass die Eigenschaften der finalen Sequenz seinen Vorstellungen möglichst nahe kommen. Auch der weitere Produktionsprozess, über die Aufspaltung in Subfragmente bis zu Oligodesign und der finalen Sequenzkontrolle wird von der Software in einer integrierten Arbeitsumgebung unterstützt.
Mit dem Sequenzanalyzer-Programm kann der mit herkömmlicher Software besonders arbeitsintensive Schritt der Sequenzkontrolle der Subfragment-Klone wesentlich erleichtert werden. Statt mehrerer Dutzend überwiegend visuell zu analysierender Sequenzen pro Subfragment muss jetzt in vielen Fällen nur noch eine Sequenz einer visuellen Endkontrolle unterzogen werden.
Insgesamt konnten die zu Beginn der Arbeit gesetzten Ziele in vollem Umfang erreicht werden. Bereits im Laufe der Programmentwicklung wurden mittlerweile über tausend synthetische Gene mit Hilfe der GeneOptimizer Suite designed und/oder ihre Herstellung softwareseitig unterstützt. Ihr Einsatzgebiet umfasst dabei ein breites Spektrum von der molekularbiologischen Grundlagenforschung über die biomedizinische Forschung (Entwicklung von DNA-Impfstoffen z.B. gegen HIV, Diagnostika etc.) bis hin zur Verbesserung biotechnologischer Produktionsprozesse (beispielsweise von Protein-Generika) und sowohl die Industrie als auch Hochschulen und öffentliche Forschungseinrichtungen zählen zum Anwenderkreis.

D.2 Ausblick

Eine Weiterentwicklung des Softwarepakets ist in erster Linie in zwei Richtungen denkbar. Dies ist zum einen natürlich die Verbesserung des Optimierungsprozesses und die Berücksichtigung weiterer Optimierungskriterien.
Analysiert man die Kodonwahl von *Escherichia coli* nicht nur hinsichlich der Frequenz der verwendeten Kodons, sondern berücksichtigt auch, welche Kodons in der

Sequenz paarweise aufeinanderfolgen, so findet man unter Berücksichtigung von weiteren Parametern wie der leicht ungleichmäßigen Verteilung von Aminosäurepaarungen, dass bestimmte Kodonpaarungen in codierenden Sequenzen unterrepräsentiert sind, andere wiederum überrepräsentiert. Des weiteren findet man, dass hochexprimierte Gene bevorzugt im Gesamttranskriptom unterrepräsentierte Kodonpaarungen aufweisen [Hatfield 1992]. Irwin et. al konnten auch im Experiment an ausgewählten Kodonpaar-Beispielen zeigen, dass sich die Schrittzeiten der Translation von unterrepräsentierten und überrepräsentierten Kodonpaarungen deutlich unterschieden [Irwin 1995]. Dies ist möglicherweise auf die unterschiedliche Kompatibilität benachbarter tRNA-Isoakzeptoren zurückzuführen. Prinzipiell könnte nun eine bei der Optimierung verwendete Gütefunktion nicht nur den Codon-Adaption-Index zur Bewertung heranziehen, sondern auch die Art der in der Testsequenz verwendeten Kodonpaarungen. Zusätzlich zur Codon-Usage-Tabelle müsste also auch eine Kodon-Paar-Tabelle hinterlegt werden. Ob dieses zusätzliche Kriterium die Expression von synthetischen Genen mit hohem CAI jedoch tatsächlich noch weiter steigern kann, könnte beispielsweise in Vergleichsexperimenten mit unterschiedlich optimierten synthetischen Genen gezeigt werden.

Wenngleich die in Kap. A.4 angesprochene Kassettenmutagenese heute an Bedeutung verloren hat, ist es dennoch des öfteren erforderlich, an einer bestimmten Stelle eine singuläre Restriktionsschnittstelle einzuführen. Diese Aufgabe lässt sich momentan mit GeneOptimizer nur umständlich lösen. Denkbar wäre hier eine Funktion, bei der der Anwender den Insertionsbereich markiert und eine Reihe in Frage kommender Restriktionsschnittstellen vorgibt. GeneOptimizer präsentiert dem Anwender dann diejenigen Schnittstellen, welche sich einerseits im gewählten Sequenzbereich durch Wahl geeigneter Kodons einführen lassen und die andererseits im Rest der Sequenz vermieden werden können. Darüber hinaus sollten in den Angaben auch Informationen darüber enthalten sein, mit welchen Kodons (häufig oder selten genutzte) die Schnittstellen eingeführt bzw. vermieden werden können. Auf diese Weise kann der Anwender unter denjenigen Schnittstellen wählen, welche zum einen unter Nutzung „guter" Kodons eingeführt werden können, die aber, ohne auf sehr schlechte Kodons zurückgreifen zu müssen, in der restlichen Sequenz vermieden werden können.
Ist die Entscheidung für eine Schnittstelle (oder auch ein anderes DNA-Motiv) gefallen, könnte dieses auf zwei Arten festgelegt werden. Zum einen könnte die Sequenz direkt an der entsprechden Stelle festgelegt und geschützt werden. Eine vorteilhaftere Methode bestünde darin, den Sitescore für die Einführung oder Vermeidung von DNA-Motiven positionsabhängig zu gestalten. Das heißt, an der Insertionsstelle wird ein positiver Sitescore vergeben, im Rest der Sequenz ein

negativer. Diese Verfahrensweise hat den Vorteil, dass bei Motiven, welche degenerierte Basenpositionen aufweisen, bei der Optimierung auch andere Kriterien weiterhin berücksichtigt werden können, da die Variationsmöglichkeiten an den degenerierten Positionen nicht eingeschränkt werden.

Eine positionsabhängige Bewertung könnte auch in Zusammenhang mit invers-komplementären Repetitionen sinnvoll sein, da insbesondere die Bildung von Haarnadelschleifen, verursacht durch nahe beieinander liegende invers-komplementäre Repetitionen, vermieden werden sollte.

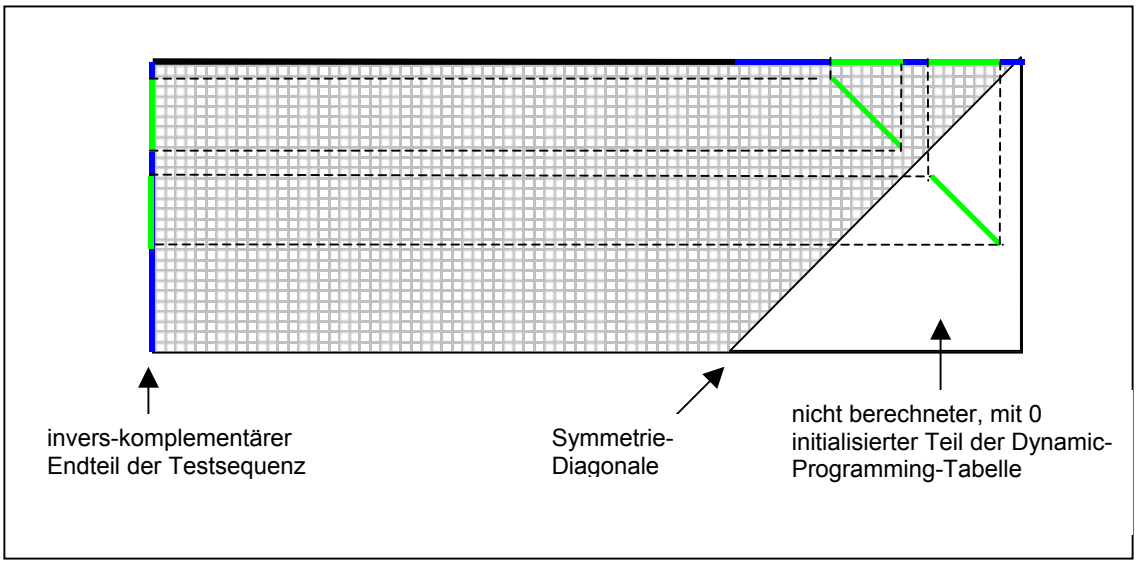


Abb. D.2-1 Erkennung von zueinander invers-komplementären Bereichen mit Hilfe des Dynamic-Programming

Will man nahe beieinander liegende zueinander invers-komplementäre Bereiche im Verhältnis zu weiter voneinander entfernten Bereichen mit einem höheren Score gewichten, so kann dies dadurch erreicht werden, dass der Match-Score in einem Bereich, der von der Symmetrie-Diagonalen und einer dazu parallelen Diagonalen begrenzt wird, höher gesetzt wird, als außerhalb dieses Bereiches.

GeneOptimizer bietet bereits die Möglichkeit, zu einer gegebenen (relativ kurzen) Sequenz homologe Abschnitte bei der Optimierung zu vermeiden. Jedoch ist die Integration einer derartigen Überprüfung in Bezug auf Homologien zu einem ganzen Wirts-Genom in den bestehenden Algorithmus offensichtlich nicht möglich, da die benötigte Rechenzeit selbst bei kleinen Genomen in inakzeptable Größenordungen vorstoßen würde. Aus zwei Gründen wäre jedoch eine derartige Überprüfung dennoch wünschenswert. Abhängig vom Einsatzgebiet des synthetischen Gens sollten ausgedehnte Homologien vermieden werden, um eine Einrekombination in das

Genom des Zielorganismus sicher zu vermeiden. Ebenfalls sollten zum Transkriptom des Organismus invers komplementäre Sequenzidentitäten von mehr als 20 Nucleotiden ausgeschlossen werden. Dies liegt im von *Science* bereits als „breakthrough of the year" [Couzin 2002] gefeierten Phänomen der RNA-Interferenz begründet. Dabei wird die Expression von Genen durch enzymatische Zerstörung der transkribierten RNA verhindert, falls diese eine exakte Sequenzübereinstimmung von mehr als 20 Nucleotiden zu einer in die Zelle eingebrachten oder dort gebildeten doppelsträngigen RNA aufweist.
Nun ist die Bildung von solch exakten (>20 Nt) oder weiter ausgedehnten aber weniger exakten Homologien jedoch eher unwahrscheinlich. In einer Erweiterung der Software könnte eine derartige Überprüfung daher nach Abschluss der Optimierung mit einer automatisierten BLAST-Suche durchgeführt werden. Etwaige Sequenzidentitäten würden dann in die Liste auszuschließender DNA-Motive aufgenommen, homologe Bereiche des Wirtsgenomes in den Homologiecheck (Projekteigenschaften->Homologiechek) einbezogen und die Optimierung erneut gestartet. Dass durch die nun erfolgende Vermeidung der Sequenzidentitäten/starker Homologien wiederum andere derartige Problemsequenzen erzeugt werden, ist nahezu ausgeschlossen. Jedoch könnte der Ablauf Optimierung->Überprüfung->erneute Optimierung mit neuen Sequenzausschlusskriterien prinzipiell so oft durchgeführt werden, bis alle Sequenzidentitäten vermieden sind bzw. die Homologie zum Genom auf ein akzeptables Maß reduziert wurde.

Durch die Verwendung einer prinzipiell beliebig komplexen Gütefunktion ist der entwickelte Algorithmus jedoch nicht auf die bereits implementierten oder oben genannten Kriterien limitiert, sondern kann auch in Zukunft flexibel an neue experimentelle Erkenntnisse und Anforderungen angepasst werden.

Mit zunehmender Automatisierung des Produktionsprozesses wird es jedoch auch erforderlich werden, die einzelnen Roboterplattformen möglichst nahtlos in die Gesamt-IT-Struktur mit der GeneOptimizer-Suite und der zugrundeliegenden Datenbank als Datenzentrale zu integrieren. Während der GeneOptimizer momentan nur die Sequenzrohdaten zur ggf. auch manuellen Weiterverarbeitung zu Verfügung stellt, werden zunehmend Steuerungsaufgaben, also beispielsweise die automatische Erstellung von Roboter-Steuerungsskripten, den Funktionsumfang von Geneoptimizer ergänzen müssen.
Konkret bedeutet dies z.B., die bereits vorhandenen Informationen zu den benötigten Oligonucleotiden (Sequenz, Synthesemaßstab, etc.) in ein Steuerprotokoll für einen automatischen Hochdurchsatz-Oligonucleotidsynthesizer zu konvertieren. Während dies im wesentlichen eine Frage des Datenformats darstellt, stellt der logisch nächste

Automatisierungsschritt, nämlich das automatisierte Zusammenpipettieren der zusammengehörigen Oligonucleotide für die Batch-Synthese der Subfragmente bereits eine anspruchsvollere Aufgabe dar. Mittel- und langfristig sollten in die IT-Struktur auch die Funktionen eines Labor-Informations-Managementsystems integriert werden. Dies bedeutet zum einen, dass jederzeit der Synthesefortschritt mit Hinweisen auf eventuell notwendige, vom Anwender auszuführende Tätigkeiten abgefragt werden kann, zum anderen können alle ausgeführten Arbeitsschritte zusammen mit Informationen bezüglich der verwendeten Chemikalien etc. protokolliert werden. Letzteres ist insbesondere in Verbindung mit dem Arbeiten unter GLP-Bedingungen wichtig. Darüber hinaus kann das System auch logistische Hilfestellung leisten, indem rechtzeitig auf die nötige Bestellung von Reagenzien etc. hingewiesen wird.

Mit der GeneOptimizer-Suite wurde die softwareseitige Grundlage für eine Hochdurchsatz-Optimierung und -Synthese artifizieller Gene gelegt. Als eines der mächtigsten Werkzeuge der Gentechnologie eröffnen synthetische Gene ungeahnte Chancen und Möglichkeiten, und es bleibt zu hoffen, dass dieses Werkzeug in wissenschaftlich und ethisch verantwortungsvoller Weise seinen Einsatz findet.

E Literaturverzeichnis

[Agrarwal 1970] Agrarwal, K.L.; Buchi, H.; Caruthers, M.H.; Gupta, N.; Khorana, H.G.; Kleppe, K.; Kumar, A.; Ohtsuka, E.; Rajbhandary U.L.; Van de Sande J.H.; Sgaramella V.; Weber, H.; Yamada, T. „Total synthesis of the gene for an alanine transfer ribonucleic acid from yeast" *Nature* 227 (1970)

[Altschul 1990] Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. „Basic local alignment search tool" *J. Mol. Biol.* 215 (3) (1990)

[Arnold 1998] Arnold, Frances H. „Enzyme engineering reaches the boiling point" *Proc. Natl. Acad. Sci. USA* 95 (1998)

[Bagga 1990] Bagga, Rajesh; Ramesh, N.; Brahmachari, K. „Supercoil-induced unusual DNA structures as transcriptional block" *Nucleic Acids Res.* 18 (11) (1990)

[Bieth 1997] Bieth, E.; Cahoreau, C.; Cholin, S.; Molinas, C.; Cerutti, M.; Rochiccioli, P.; Devauchelle, G.; Tauber, M. „Human growth hormone receptor: cloning and expression of the full-length complementary DNA after site-directed inactivation of a cryptic bacterial promoter" *Gene* 194 (1997)

[Bleasby 2000] Bleasby, Alan „Programming EMBOSS Applications" *http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Doc/Develop/program.html* (2000)

[Borer 1974] Borer, Philip N.; Dengler, Barbara; Tinoco, Ignacio Jr. „Stability of Ribonucleic acid Double-stranded Helices" *J. Mol. Biol.* 86 (1974)

[Brahmachari 1991] Brahmachari S.K.; Sarkar, P.S.; Balagurumoorthy P.; Burma, P.K.; Bagga, R. „Synthetic gene design to investigate the role of cis-acting DNA-structural elements in regulation of gene Expression in vivo" *Nucleic Acids symposium series* 24 (1991)

[Breslauer 1986] Breslauer, Kenneth J.; Frank, Ronald; Blöcker, Helmut; Marky, Luis A. „Predicting DNA duplex stability from the base sequence" *Proc. Natl. Acad. Sci.* 83 (1986)

[Chen 1993] Chen C.; Mao, J.; Zhang M.; Dai, J. „Combination of DNA single strand synthesis with PCR to construct mung bean trypsin inhibitor gene" *Chinese Journal of Biotechnology* 9(1) (1993)

[Ciccarelli 1991] Ciccarelli, Richard B.; Gunyuzlu, Paul; Huang, James; Scott, Charles; Oakes, T. Fred „Construction of synthetic genes using PCR after automated DNA synthesis of their entire top and bottom strands" *Nucleic Acids Res.* 19 (21) (1991)

[Couzin 2002] Couzin J. „Breakthrough of the year. Small RNAs make big splash" *Science* 298 (5602) (2002)

[Dillon 2000] Dillon, Patrick J.; Rosen, Craig A. „Construction of synthetic genes by polymerase chain reaction" *Nucleic Acid Protocols Handbook, Humana Press Inc., Totowa* (2000)

[Dong 1996] Dong, H.; Nilsson, L.; Kurland, C.G. „Co-variation of tRNA Abundance and Codon Usage in Escherichia coli at Different Growth Rates" *J. Mol. Biol.* 260 (1996)

[Goldman 1995] Goldman, Emanuel; Rosenberg, Alan H.; Zubay, Geoffrey; Studier, F. William „Consecutive Low-usage Leucine Codons Block Translation Only When Near the 5' End of a Message in Escherichia coli" *J. Mol. Biol.* 245 (1995)

[Griswold 2003] Griswold, Karl E.; Mahmood, Nadir A.; Iverson, Brent L.; Georgiou, George „Effects of codon usage versus putative 5'-mRNA structure on the expression of Fusarium solani cutinase in the Escherichia coli cytoplasm" *Protein Expression & Purification* 27 (2003)

[Gusfield 1999] Gusfield, Dan *Algorithms on strings, trees and sequences - computer science and computational biology* (Cambridge, Cambridge Univ. Press, 1999)

[Hardy 1997] Hardy, Paul; Waterman, Michael S. „The Sequence Alignment Software Library at USC" *http://www-hto.usc.edu/software/seqaln/doc/seqalndoc.ps* (1997)

[Hatfield 1992] Hatfield, G. Wesley; Gutman, George A. „Codon Pair Utilization" *US Patent 5,082,767* (1992)

[Hoover 2002] Hoover, David M.; Lubkowski, Jacek „DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis" *Nucleic Acids Res.* 30 (10) (2002)

[Hui 1999] Hui, Vernon W. „Microsoft Beefs up VBScript with Regular Expressions" *msdn online; http://msdn.microsoft.com/workshop/languages/clinic/scripting051099.asp* (1999)

[Ikemura 1982] Ikemura, Toshimichi „Correlation between the abundance of yeast transfer RNAs and the occurrence of the respective codons in protein genes. Differences in synonymous codon choice patterns of yeast and E. coli with reference to the abundance of isoaccepting transfer RNAs" *J. Mol. Biol.* 158(4) (1982)

[Irwin 1995] Irwin, Becky; Heck, Denis J.; Hatfield, G. Wesley „Codon Pair Utilization Biases Influence Translational Elongation Step Times" *Journal of Biological Chemistry* 270 (39) (1995)

[Itakura 1977] Itakura, K.; Hirose, T.; Crea, R.; Riggs, A.D.; Heyneker, H.L.; Bolivar, F.; Boyer, H.W. „Expression in Eschrichia coli of a chemically synthesized gene for the hormone somatostatin" *Science* 198(4321) (1977)

[Jerala 1988] Jerala, Roman; Turk, Vito „Regen: program for designing gene assembly" *Nucleic Acids Res.* 16 (5) (1988)

[Kaderali 2001] Kaderali, Lars „Selecting Target Specific Probes for DNA Arrays" *Diplomarbeit* (2001)

[Kane 1995] Kane, James F. „Effects of rare codon clusters on high-level expression of heterologous proteins in Escherichia coli" *Current Opinion in Biotechnology* 6 (1995)

[Knappik 2000] Knappik, A.; Liming, G.; Honegger, A.; Pack, P.; Fischer, M.; Wellnhofer, G.; Hoess, A.; Wölle, J.; Plückthunn, A.; Virnekäs, B. „Fully Synthetic Human Combinatorial Antibody Libraries (HuCAL) Based on Modular Consensus Frameworks and CDRs Randomized with Trinucleotides" *J. Mol. Biol.* 296 (2000)

[Köster 1978] Köster, Hubert; Blöcker, Helmut; Frank, Ronald; Geussenhainer, Stefan; Kaiser, Wolfgang „Synthese eines Strukturgens für ds Peptidhormon Angiotensin II, Teil 1 - Zielsetzung und Synthesestrategie" *Justus Liebigs Annalen der Chemie* 6 (1978)

[Lee 2002] Lee, Sang-Gu; Kim, Dae-You; Hyun, Byung-Hwa; Bae, Yong-Soo „Novel Design Architecture for Genetic Stability of Recombinant Poliovirus: the Manipulation of G/C Contents and their Distribution Patterns increases the Genetic

Stability of Inserts in a Poliovirus-Based RPS-Vax Vector System" *Journal of Virology* 76 (4) (2002)

[Libertini 1992] Libertini, Giacinto; Di Donato, Alberto „Computer aided gene design" *Protein Eng.* 5 (8) (1992)

[Liu 2000] Liu, Margaret A.; Ulmer, Jeffrey B. „Gene-Based Vaccines" *Molecular Therapy* 1 (6) (2000)

[Makarova 1992] Makarova, K.S.; Mazin, A.V.; Wolf, Yu I.; Soloviev, V.V. „DIROM: an experimental design interactive system for directed mutagenesisi and nucleic acids engineering" *Comp. Appl. Biosci.* 8 (5) (1992)

[Mathur 1991] Mathur, M.; Tuli, R. „Analysis of Codon Usage in Genes for Nitrogen Fixation from Phylogenetically Diverse Diazotrophs" *Journal of Molecular Evolution* 32 (1991)

[Müller 1986] Müller, Uwe R.; Turnage, Michael A. „Insertions of Palindromic DNA Sequences into the J-F Intercistronic Region of Bacteriophage PhiX174 Interfere with Normal Phage Growth" *J. Mol. Biol.* 189 (1986)

[Nakamura 2000] Nakamura, Y.; Gojobori, T; Ikemura, T. „Codon usage tabulated from the international DNA sequence databases: status for the year 2000" *Nucl. Acids Res.* 28 (2000)

[NCBI 2001] NCBI „Qblast's URL API. User's Guide" *http://www.ncbi.nlm.nih.gov/BLAST/Doc/urlapi.pdf* (2001)

[Ochagavia 1992] Ochagavia, M.; Jiménez, V.; Fernández de Cossio; Suárez, A.; Bringas, R.; Ricardo, R. „SYNSOS: Paquete de programas de ayuda en el diseno de genes" *Biotecnologia Aplicada* 9 (1) (1992)

[Petri 1989] Petri, I.; Troebner, W.; Guehrs, K.H., Behnke, D. „Computer aided gene design" *studia biophysica* 129 (2-3) (1989)

[Presnell 1988] Presnell, Scott R; Brenner, Steven A „The Design of synthetic Genes" *Nucleic Acids Res.* 16(5) (1988)

[Quandt 1995] Quandt, Kerstin; Frech, Kornelie; Karas, Holger; Wingender, Edgar; Werner, Thomas „MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data" *Nucleic Acids Res.* 23 (23) (1995)

[Raghava 1994] Raghava, G.P.S.; Sahni, Girish; „GMAP: a multi-purpose computer program to aid synthetic gene design, cassette mutagenesis and the introduction of potential restriction sites into DNA sequences" *BioTechniques* 16 (6) (1994)

[Rosenberg 1993] Rosenberg, Alan H.; Goldman, Emanuel; Dunn, John J.; Studier, F. William; Zubay, Geoffrey „Effects of Consecutive AGG Codons on Translation in Escherichia coli, Demonstrated with a Versatile Codon Test System" *Journal of Bacteriology* 175 (3) (1993)

[Sambrook 2001] Sambrook, Joseph; Russell, David W. *Molecular cloning: a laboratory manual* (Cold Spring Harbor Laboratory Press, 2001)

[SantaLucia 1996] SantaLucia, John Jr; Allawi, Hatim T.; Seneviratne, Ananda P. „Improved Nearest-Neighbour Parameters for Predicting DNA Duplex Stability" *Biochemistry* 35 (1996)

[Schatz 2000] Schatz, Octavian „Solid-phase synthesis of DNA-fragments using ligation and enzymic restriction techniques" *PCT Int. Appl WO 0075368* (2000)

[Sharp 1987] Sharp, P.M.; Li, W. H. „The codon Adaptation Index-a measure of directional synonymous codon usage bias, and its potential applications" *Nucleic Acids Res.* 15 (3) (1987)

[Smith 1996] Smith, David W.E. „Problems of Translating Heterologous Genes in Expression Systems: The Role of tRNA" *Biotechnol. Prog.* 12 (1996)

[Staahl 1993] Staahl, S.; Hansson, M.; Ahlborg, N.; Nguyen, T.N.; Liljeqvist, S.; Lundeberg, J.; Uhlen, M. „Solid-phase gene assembly of constructs derived from the Plasmodium falciparum malaria blood-stage antigen Ag332" *BioTechniques* 14(3) (1993)

[Thomson 2001] Thomson, Scott A.; Ramshaw, Ian A. „Synthetic Peptides and uses therefore" *PCT Anmeldung WO 01/90197 A1* (2001)

[Tibbetts 1995] Tibbetts, Clark „Raw Data File Formats, and the Digital and Analog Raw Data Streams of the ABI PRISM 377 DNA Sequencer; A preliminary technical examination"
http://www.cs.cmu.edu/afs/cs/project/genome/ftp/other/377_Raw_Data.ps (1995)

[Tylor 2001] Tylor, Sean V.; Kast, Peter; Hilvert, Donald „Investigating and Engineering Enzymes by Genetic Selection" *Angew. Chem. Int. Ed.* 40 (2001)

[Wagner 2000] Wagner, Ralf; Graf, Marcus; Bieler, Kurt; Wolf, Hans; Grunwald, Thomas; Foley, P.; Überla, Klaus „Rev-Independent Expression of Synthetic gag–pol Genes of Human Immunodeficiency Virus Type 1 and Simian Immunodeficiency Virus: Implications for the Safety of Lentiviral Vectors" *Human Gene Therapy* 11 (2000)

[Waterman 1987] Waterman, M.S.; Eggert, M. „A New Algorithm for Best Subsequence Alignments with Appliction to tRNA-rRNA Comparisons" *J. Mol. Biol* 197 (1987)

[Waterman 2000] Waterman, Michael S. *Introduction to computational biology - maps, sequences and genomes* (CRC Press, Boca Raton, 2000)

[Weiner 1989] Weiner, M. P.; Scheraga, H. A. „A set of Macintosh computer programs for the design and analysis of synthetic genes" *Comput. Appl. Biosci.* 5 (3) (1989)

[Wolff 1990] Wolff, J.A.; Malone, R.W.; Williams, P.; Chong, W.; Acsadi, G.; Jani, A.; Felgner, P.L. „Direct gene transfer into mouse muscle in vivo" *Science* 247 (1990)

F Anhang

Zur Erläuterung siehe Kap. B.2.3.9

KDS-Startposition 1 bei Aminosäure **1 E**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
GAACAGTTC GAACAGTTC	92	5	0	0,0	G | G	87,0
GAACAGTTT GAACAGTTT	100	19	0	0,0	TT || TT	81,0
GAGCAGTTT GAGCAGTTT	82	5	0	0,0	AG || AG	77,0
GAGCAGTTC GAGCAGTTC	73	5	0	0,0	AG || AG	68,0
GAACAATTC GAACAATTC	76	19	0	0,0	AA || AA	57,0
GAGCAATTC GAGCAATTC	58	5	0	0,0	G | G	53,0
GAACAATTT GAACAATTT	85	38	0	0,0	AA || AA	47,0
GAGCAATTT GAGCAATTT	66	19	0	0,0	TT || TT	47,0

KDS-Startposition 4 bei Aminosäure **2 Q**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
CAGTTCATC GAACAGTTCATC	86	8	0	0,0	CA || CA	78,0
CAGTTTATC GAACAGTTTATC	94	19	0	0,0	TT || TT	75,0
CAGTTCATT GAACAGTTCATT	92	19	0	0,0	CA || CA	73,0
CAGTTTATT GAACAGTTTATT	100	33	0	0,0	TT || TT	67,0
CAATTCATC GAACAATTCATC	70	19	0	0,0	AA || AA	51,0
CAATTTATC GAACAATTTATC	79	33	0	0,0	AA || AA	46,0
CAGTTCATA GAACAGTTCATA	63	19	0	0,0	CA || CA	44,0
CAATTCATT GAACAATTCATT	76	33	0	0,0	ATT ||| ATT	43,0
CAGTTTATA GAACAGTTTATA	71	33	0	0,0	TT || TT	38,0

CAATTTATT GAACAATTTATT	85	48	0	0,0	ATT ||| ATT	37,0
CAATTCATA GAACAATTCATA	48	33	0	0,0	AA || AA	15,0
CAATTTATA GAACAATTTATA	56	48	0	0,0	AA || AA	8,0

KDS-Startposition 7 bei Aminosäure **3 F**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
TTCATCATC GAACAGTTCATCATC	80	10	0	0,0	TCATC ||||| TCATC	70,0
TTTATCATC GAACAGTTTATCATC	88	19	0	0,0	ATC ||| ATC	69,0
TTCATTATC GAACAGTTCATTATC	86	19	0	0,0	CA || CA	67,0
TTCATCATT GAACAGTTCATCATT	86	19	0	0,0	TCAT |||| TCAT	67,0
TTTATTATC GAACAGTTTATTATC	94	30	0	0,0	TTAT |||| TTAT	64,0
TTTATCATT GAACAGTTTATCATT	94	30	0	0,0	CA || CA	64,0
TTCATTATT GAACAGTTCATTATT	92	30	0	0,0	ATT ||| ATT	62,0
TTTATTATT GAACAGTTTATTATT	100	42	0	0,0	TTATT ||||| TTATT	58,0
TTCATCATA GAACAGTTCATCATA	57	19	0	0,0	TCAT |||| TCAT	38,0
TTCATAATC GAACAGTTCATAATC	57	19	0	0,0	AA || AA	38,0
TTTATCATA GAACAGTTTATCATA	65	30	0	0,0	CA || CA	35,0
TTTATAATC GAACAGTTTATAATC	65	30	0	0,0	AA || AA	35,0
TTCATTATA GAACAGTTCATTATA	63	30	0	0,0	CA || CA	33,0
TTCATAATT GAACAGTTCATAATT	63	30	0	0,0	AA || AA	33,0
TTTATTATA GAACAGTTTATTATA	71	42	0	0,0	TTAT |||| TTAT	29,0
TTTATAATT GAACAGTTTATAATT	71	42	0	0,0	AA || AA	29,0
TTCATAATA GAACAGTTCATAATA	34	30	0	0,0	ATA ||| ATA	4,0
TTTATAATA GAACAGTTTATAATA	43	42	0	0,0	ATA ||| ATA	1,0

KDS-Startposition 10 bei Aminosäure **4 I**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATCATCAAA GAACAGTTCATCATCAAA	88	19	0	0,0	TCATCA |||||| TCATCA	69,0
ATTATCAAA GAACAGTTCATTATCAAA	94	28	0	0,0	TCA ||| TCA	66,0
ATCATTAAA GAACAGTTCATCATTAAA	94	28	0	0,0	TCAT |||| TCAT	66,0
ATTATTAAA GAACAGTTCATTATTAAA	100	38	0	0,0	ATTA |||| ATTA	62,0
ATCATCAAG GAACAGTTCATCATCAAG	65	11	0	0,0	TCATCA |||||| TCATCA	54,0
ATTATCAAG GAACAGTTCATTATCAAG	71	19	0	0,0	TCA ||| TCA	52,0
ATCATTAAG GAACAGTTCATCATTAAG	71	19	0	0,0	TCAT |||| TCAT	52,0
ATTATTAAG GAACAGTTCATTATTAAG	77	28	0	0,0	ATTA |||| ATTA	49,0
ATCATAAAA GAACAGTTCATCATAAAA	65	28	0	0,0	TCAT |||| TCAT	37,0
ATAATCAAA GAACAGTTCATAATCAAA	65	28	0	0,0	TCA ||| TCA	37,0
ATTATAAAA GAACAGTTCATTATAAAA	71	38	0	0,0	AAA ||| AAA	33,0
ATAATTAAA GAACAGTTCATAATTAAA	71	38	0	0,0	TAA ||| TAA	33,0
ATCATAAAG GAACAGTTCATCATAAAG	43	19	0	0,0	TCAT |||| TCAT	24,0
ATAATCAAG GAACAGTTCATAATCAAG	43	19	0	0,0	TCA ||| TCA	24,0
ATTATAAAG GAACAGTTCATTATAAAG	49	28	0	0,0	AA || AA	21,0
ATAATTAAG GAACAGTTCATAATTAAG	49	28	0	0,0	TAA ||| TAA	21,0
ATAATAAAA GAACAGTTCATAATAAAA	43	38	0	0,0	ATAA |||| ATAA	5,0
ATAATAAAG GAACAGTTCATAATAAAG	20	28	0	0,0	ATAA |||| ATAA	-8,0

KDS-Startposition 13 bei Aminosäure **5 I**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATCAAAAAC GAACAGTTCATCATCAAAAAC	94	19	0	0,0	TCATCA |||||| TCATCA	75,0
ATTAAAAAC GAACAGTTCATCATTAAAAAC	100	27	0	0,0	TCAT |||| TCAT	73,0

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATCAAAAAT GAACAGTTCATCATCAAAAAT	88	27	0	0,0	TCATCA |||||| TCATCA	61,0
ATTAAAAAT GAACAGTTCATCATTAAAAAT	94	35	0	0,0	TCAT |||| TCAT	59,0
ATTAAGAAC GAACAGTTCATCATTAAGAAC	77	19	0	0,0	GAAC |||| GAAC	58,0
ATCAAGAAC GAACAGTTCATCATCAAGAAC	71	13	0	0,0	TCATCA |||||| TCATCA	58,0
ATCAAGAAT GAACAGTTCATCATCAAGAAT	65	19	0	0,0	TCATCA |||||| TCATCA	46,0
ATTAAGAAT GAACAGTTCATCATTAAGAAT	71	27	0	0,0	TCAT |||| TCAT	44,0
ATAAAAAAC GAACAGTTCATCATAAAAAAC	71	27	0	0,0	TCAT-A-AAAAA |||| | ||||| TCATCATAAAAA	44,0
ATAAAAAAT GAACAGTTCATCATAAAAAAT	65	35	0	0,0	TCAT-A-AAAAA |||| | ||||| TCATCATAAAAA	30,0
ATAAAGAAC GAACAGTTCATCATAAAGAAC	49	19	0	0,0	GAAC |||| GAAC	30,0
ATAAAGAAT GAACAGTTCATCATAAAGAAT	43	27	0	0,0	TCAT |||| TCAT	16,0

KDS-Startposition 16 bei Aminosäure **6 K**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
AAAAATATG GAACAGTTCATCATCAAAAATATG	94	26	0	0,0	TCATCA |||||| TCATCA	68,0
AAGAATATG GAACAGTTCATCATCAAGAATATG	71	19	0	0,0	TCATCA |||||| TCATCA	52,0
AAAAACATG GAACAGTTCATCATCAAAAACATG	100	19	200000	0,0	TCATCA |||||| TCATCA	919,0
AAGAACATG GAACAGTTCATCATCAAGAACATG	77	13	200000	0,0	TCATCA |||||| TCATCA	936,0

KDS-Startposition 19 bei Aminosäure **7 N**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
AATATGTTT GAACAGTTCATCATCAAAAATATGTTT	94	35	0	0,0	TCATCA |||||| TCATCA	59,0
AATATGTTC GAACAGTTCATCATCAAAAATATGTTC	86	28	0	0,0	TCATCA |||||| TCATCA	58,0
AACATGTTT GAACAGTTCATCATCAAAAACATGTTT	100	28	200000	0,0	TCATCA |||||| TCATCA	928,0
AACATGTTC GAACAGTTCATCATCAAAAACATGTTC	92	21	200000	0,0	AACATGTTC |||| |||| AACA-GTTC	929,0

KDS-Startposition 22 bei Aminosäure **8 M**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATGTTTATC GAACAGTTCATCATCAAAAATATGTTTATC	94	35	0	0,0	TCATCA |||||| TCATCA	59,0
ATGTTTATT GAACAGTTCATCATCAAAAATATGTTTATT	100	42	0	0,0	TCATCA |||||| TCATCA	58,0
ATGTTCATT GAACAGTTCATCATCAAAAATATGTTCATT	92	35	0	0,0	GTTCAT |||||| GTTCAT	57,0
ATGTTCATC GAACAGTTCATCATCAAAAATATGTTCATC	86	28	0	12,5	GTTCATC ||||||| GTTCATC	45,0
ATGTTTATA GAACAGTTCATCATCAAAAATATGTTTATA	71	42	0	0,0	TCATCA |||||| TCATCA	29,0
ATGTTCATA GAACAGTTCATCATCAAAAATATGTTCATA	63	35	0	0,0	GTTCAT |||||| GTTCAT	28,0

KDS-Startposition 25 bei Aminosäure **9 F**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
TTTATTATC GAACAGTTCATCATCAAAAATATGTTTATTATC	94	42	0	0,0	TCATCA |||||| TCATCA	52,0
TTTATCATT GAACAGTTCATCATCAAAAATATGTTTATCATT	94	42	0	0,0	TCATCA |||||| TCATCA	52,0
TTCATTATT GAACAGTTCATCATCAAAAATATGTTCATTATT	92	42	0	0,0	GTTCAT |||||| GTTCAT	50,0
TTTATCATC GAACAGTTCATCATCAAAAATATGTTTATCATC	88	35	0	12,5	GTTTATCATC ||| ||||| GTTCATCATC	40,0
TTTATTATT GAACAGTTCATCATCAAAAATATGTTTATTATT	100	49	0	12,5	TCATCA--AAAATATGTTTATTATT ||||| |||| || | | ||||| TCATCATCAAAA-ATATGT-TTATT	38,0
TTCATTATC GAACAGTTCATCATCAAAAATATGTTCATTATC	86	35	0	12,5	GTTCATTATC |||||| ||| GTTCATCATC	38,0
TTCATCATT GAACAGTTCATCATCAAAAATATGTTCATCATT	86	35	0	17,4	GTTCATCAT ||||||||| GTTCATCAT	34,0
TTCATCATC GAACAGTTCATCATCAAAAATATGTTCATCATC	80	28	0	20,0	GTTCATCATC ||||||||||| GTTCATCATC	32,0
TTTATCATA GAACAGTTCATCATCAAAAATATGTTTATCATA	65	42	0	0,0	TCATCA |||||| TCATCA	23,0
TTTATAATC GAACAGTTCATCATCAAAAATATGTTTATAATC	65	42	0	0,0	TCATCA |||||| TCATCA	23,0
TTTATTATA GAACAGTTCATCATCAAAAATATGTTTATTATA	71	49	0	0,0	TCATCA |||||| TCATCA	22,0
TTTATAATT GAACAGTTCATCATCAAAAATATGTTTATAATT	71	49	0	0,0	TCATCA |||||| TCATCA	22,0
TTCATAATT GAACAGTTCATCATCAAAAATATGTTCATAATT	63	42	0	0,0	GTTCAT |||||| GTTCAT	21,0
TTCATTATA GAACAGTTCATCATCAAAAATATGTTCATTATA	63	42	0	0,0	GTTCAT |||||| GTTCAT	21,0

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
TTCATAATC GAACAGTTCATCATCAAAAATATGTTCATAATC	57	35	0	12,5	GTTCATAATC |||||| ||| GTTCATCATC	9,0
TTCATCATA GAACAGTTCATCATCAAAAATATGTTCATCATA	57	35	0	17,4	GTTCATCAT ||||||||| GTTCATCAT	5,0
TTTATAATA GAACAGTTCATCATCAAAAATATGTTTATAATA	43	49	0	0,0	TCATCA |||||| TCATCA	-6,0
TTCATAATA GAACAGTTCATCATCAAAAATATGTTCATAATA	34	42	0	0,0	GTTCAT |||||| GTTCAT	-8,0

KDS-Startposition 28 bei Aminosäure **10 I**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATTATCAAA GAACAGTTCATCATCAAAAATATGTTTATTATCAAA	94	49	0	12,5	GTTTATTATCAAA ||| || |||||| GTTCATCATCAAA	32,0
ATCATTAAA GAACAGTTCATCATCAAAAATATGTTTATCATTAAA	94	49	0	12,5	GTTTATCATTAAA ||| |||| ||| GTTCATCATCAAA	32,0
ATTATCAAG GAACAGTTCATCATCAAAAATATGTTTATTATCAAG	71	42	0	0,0	TCATCA |||||| TCATCA	29,0
ATCATTAAG GAACAGTTCATCATCAAAAATATGTTTATCATTAAG	71	42	0	0,0	TCATCA |||||| TCATCA	29,0
ATTATTAAA GAACAGTTCATCATCAAAAATATGTTTATTATTAAA	100	57	0	14,9	TCATCA--AAAATATGTTTATTATTA ||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	28,0
ATCATCAAA GAACAGTTCATCATCAAAAATATGTTTATCATCAAA	88	42	0	20,0	GTTTATCATCAAA ||| |||||||| GTTCATCATCAAA	26,0
ATTATAAAA GAACAGTTCATCATCAAAAATATGTTTATTATAAAA	71	57	0	0,0	TCATCA |||||| TCATCA	14,0
ATAATTAAA GAACAGTTCATCATCAAAAATATGTTTATAATTAAA	71	57	0	0,0	TCATCA |||||| TCATCA	14,0
ATTATTAAG GAACAGTTCATCATCAAAAATATGTTTATTATTAAG	77	49	0	14,9	TCATCA--AAAATATGTTTATTATTA |||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	13,0
ATCATCAAG GAACAGTTCATCATCAAAAATATGTTTATCATCAAG	65	35	0	17,4	GTTTATCATCAA ||| |||||||| GTTCATCATCAA	13,0
ATAATCAAA GAACAGTTCATCATCAAAAATATGTTTATAATCAAA	65	49	0	12,5	GTTTATAATCAAA ||| || |||||| GTTCATCATCAAA	3,0
ATCATAAAA GAACAGTTCATCATCAAAAATATGTTTATCATAAAA	65	49	0	14,9	GTTTATCAT-AAAA ||| |||| |||| GTTCATCATCAAAA	1,0
ATAATCAAG GAACAGTTCATCATCAAAAATATGTTTATAATCAAG	43	42	0	0,0	TCATCA |||||| TCATCA	1,0
ATTATAAAG GAACAGTTCATCATCAAAAATATGTTTATTATAAAG	49	49	0	0,0	TCATCA |||||| TCATCA	0,0
ATAATTAAG GAACAGTTCATCATCAAAAATATGTTTATAATTAAG	49	49	0	0,0	TCATCA |||||| TCATCA	0,0
ATCATAAAG GAACAGTTCATCATCAAAAATATGTTTATCATAAAG	43	42	0	12,5	GTTTATCAT-AAA ||| ||||| ||| GTTCATCATCAAA	-12,0
ATAATAAAA GAACAGTTCATCATCAAAAATATGTTTATAATAAAA	43	57	0	0,0	TCATCA |||||| TCATCA	-14,0
ATAATAAAG GAACAGTTCATCATCAAAAATATGTTTATAATAAAG	20	49	0	0,0	TCATCA |||||| TCATCA	-29,0

KDS-Startposition 31 bei Aminosäure **11 I**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
ATCAAGAAC GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAAC	71	42	0	0,0	TCATCA |||||| TCATCA	29,0
ATTAAAAAC GAACAGTTCATCATCAAAAATATGTTTATTATTAAAAAC	100	57	0	14,9	TCATCA--AAAATATGTTTATTATTA |||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	28,0
ATCAAAAAC GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAAC	94	49	0	17,4	GTTTATTATCAAAAA ||| || |||||||| GTTCATCATCAAAAA	28,0
ATTAAAAAT GAACAGTTCATCATCAAAAATATGTTTATTATTAAAAAT	94	64	0	14,9	TCATCA--AAAATATGTTTATTATTA |||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	15,0
ATTAAGAAC GAACAGTTCATCATCAAAAATATGTTTATTATTAAGAAC	77	49	0	14,9	TCATCA--AAAATATGTTTATTATTA |||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	13,0
ATCAAAAAT GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAAT	88	57	0	20,0	GTTTATTATCAAAAAT ||| || |||||||| GTTCATCATCAAAAAT	11,0
ATCAAGAAT GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAAT	65	49	0	12,5	GTTTATTATCAAGAAT ||| || |||| ||| GTTCATCATCAAAAAT	3,0
ATAAAGAAC GAACAGTTCATCATCAAAAATATGTTTATTATAAAGAAC	49	49	0	0,0	TCATCA |||||| TCATCA	0,0
ATTAAGAAT GAACAGTTCATCATCAAAAATATGTTTATTATTAAGAAT	71	57	0	14,9	TCATCA--AAAATATGTTTATTATTA |||||| |||| || | | |||||| TCATCATCAAAA-ATATGT-TTATTA	-1,0
ATAAAAAAC GAACAGTTCATCATCAAAAATATGTTTATTATAAAAAAC	71	57	0	14,9	TCATCA--AAAATATGTTTATTA-TA-AAAAA |||||| |||| || | | ||| || ||||| TCATCATCAAAA-ATATGT-TTATTATAAAAA	-1,0
ATAAAAAAT GAACAGTTCATCATCAAAAATATGTTTATTATAAAAAAT	65	64	0	14,9	TCATCA--AAAATATGTTTATTA-TA-AAAAA |||||| |||| || | | ||| || ||||| TCATCATCAAAA-ATATGT-TTATTATAAAAA	-14,0
ATAAAGAAT GAACAGTTCATCATCAAAAATATGTTTATTATAAAGAAT	43	57	0	0,0	TCATCA |||||| TCATCA	-14,0

KDS-Startposition 34 bei Aminosäure **12 K**

KDS Testsequenz	CU	GC	Site	Rep	Alignment	Gesamt-Score
AAGAACGCG GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAACGCG	77	28	0	0,0	TCATCA |||||| TCATCA	49,0
AAAAACGCG GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAACGCG	100	35	0	17,4	GTTTATTATCAAAAA ||| || |||||||| GTTCATCATCAAAAA	48,0
AAGAACGCC GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAACGCC	69	28	0	0,0	TCATCA |||||| TCATCA	41,0
AAAAACGCC GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAACGCC	92	35	0	17,4	GTTTATTATCAAAAA ||| || |||||||| GTTCATCATCAAAAA	40,0
AAAAATGCG GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAATGCG	94	42	0	20,0	GTTTATTATCAAAAAT ||| || |||||||| GTTCATCATCAAAAAT	32,0
AAGAACGCA GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAACGCA	63	35	0	0,0	TCATCA |||||| TCATCA	28,0
AAAAACGCA GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAACGCA	86	42	0	17,4	GTTTATTATCAAAAA ||| || |||||||| GTTCATCATCAAAAA	27,0
AAAAATGCC GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAATGCC	86	42	0	20,0	GTTTATTATCAAAAAT ||| || |||||||| GTTCATCATCAAAAAT	24,0

AAGAACGCT	59	35	0	0,0	TCATCA |||||| TCATCA	24,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAACGCT						
AAGAATGCG	71	35	0	12,5	GTTTATTATCAAGAAT ||| || |||| ||| GTTCATCATCAAAAAT	23,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAATGCG						
AAAAACGCT	81	42	0	17,4	GTTTATTATCAAAAA ||| || ||||||| GTTCATCATCAAAAA	22,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAACGCT						
AAGAATGCC	63	35	0	12,5	GTTTATTATCAAGAAT ||| || |||| ||| GTTCATCATCAAAAAT	15,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAATGCC						
AAAAATGCA	80	49	0	20,0	GTTTATTATCAAAAAT ||| || |||||||| GTTCATCATCAAAAAT	11,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAATGCA						
AAAAATGCT	75	49	0	20,0	GTTTATTATCAAAAAT ||| || |||||||| GTTCATCATCAAAAAT	6,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAAAATGCT						
AAGAATGCA	57	42	0	12,5	GTTTATTATCAAGAAT ||| || |||| ||| GTTCATCATCAAAAAT	2,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAATGCA						
AAGAATGCT	53	42	0	12,5	GTTTATTATCAAGAAT ||| || |||| ||| GTTCATCATCAAAAAT	-2,0
GAACAGTTCATCATCAAAAATATGTTTATTATCAAGAATGCT						

Danksagung

Mein herzlicher Dank gebührt

PD Dr. Jörg Enderlein, ohne dessen Altruismus, Flexibilität und ständige wohlwollende Anerkennung und Unterstützung diese Arbeit nicht entstanden wäre. Ich bin ihm immer in Dank und Freundschaft verbunden.

Prof. Dr. Dr. Ralf Wagner für die Möglichkeit, diese Arbeit in einem spannenden und anwendungsorientierten Umfeld durchführen zu können und für seine freundschaftliche Unterstützung.

Dr. Marcus „Murphy" Graf für die freundschaftliche Zusammenarbeit und dass er die vielen „Bugs" (na ja, sooo viele warns ja dann doch nicht, oder?) mit Humor und Verständnis ertragen hat. Ohne seine zahllosen Wünsche und Anregungen wäre die Software nur ein Bruchteil von dem, was sie heute darstellt.

Der Geneart GmbH für die großzügige finanzielle Unterstützung.

Christian Ehl, der die o.g. Unterstützung, zusammen mit Ralf Wagner, großmütig bewilligt hat.

Meinen Kollegen von Geneart, die mir zu meiner „beruflichen Familie" geworden sind, für die freundliche Aufnahme und ihre herzliche Unterstützung.

Dr. Francesco Pampaloni für die vielen anregenden Gespräche, seine große Anteilnahme an meiner Arbeit und nicht zuletzt für seine Freundschaft, von der ich hoffe, dass sie geographische Entfernungen überwindet und noch lange Bestand hat. Lieber Francesco, ich wünsche Dir alles Gute für Deine private und berufliche Zukunft!

Dr. Martin Böhmer für die vielen ermunternden und anregenden Diskussionen in der Raucherecke.

Prof. Dr. Werner Kunz und Prof. Dr. Steinem, die mir großzügig Räumlichkeiten zur Mitarbeit an den Forschungsprojekten „Fluorophore" und „Vereinzelung" zur Verfügung gestellt haben.

Allen nicht namentlich genannten Helfern, die zum Gelingen dieser Arbeit beigetragen haben.

Insbesondere meinen Eltern und meiner Familie für ihre fortwährende Unterstützung.